

Android项目实战

智能农业移动管理系统开发

主 编 ◎ 张荣超 沈 阳

副主编 ◎ 杨种学 陈 静 包依勤 王小正

参 编 ◎ 赵 烽 王绪峰 邓所春 张喜生 胡光永 王道乾 任洪亮

毕传林 毋建军 赵 彦 张成叔 刘 学 王金兰 张宗国

徐义晗 伏玉琛 陈常晖 谢景明 周 颖

主 审 ◎ 温 涛

Intelligent
Agricultural
Mobile
Management
System
Development



Android 项目实战——智能农业移动管理系统开发/张荣超,沈阳主编.

—大连:东软电子出版社,2015.9

ISBN 978-7-89436-322-0

策划编辑: 卫果曼

责任编辑: 朱 娜

光盘开发: 马李昕

装帧设计: 董俐莉

出版/发行: 东软电子出版社

地 址: 大连市软件园路 8 号

邮 编: 116023

电话/传真: 0411-84835089

网 址: <http://www.neubooks.com>

电子邮箱: nep@neusoft.edu.cn

出版时间: 2015 年 9 月

印制时间: 2015 年 9 月第 1 次印制

字 数: 336 千字

印 制 者: 大连华录影音实业有限公司

前　言

在我国，伴随着 4G 时代（第 4 代移动通信）的开启，又一次为日渐崛起甚至井喷的移动互联网注入了巨大的能量，而与之对应的技术、平台、商业模式、应用、服务等也正不断地发展和转型。在这样的时代，无论是互联网企业还是传统行业，为了持续响应新的客户需求，占领新兴市场，增强战略执行力，积极构建自身在互联网的实力或向互联网转型，都需要大量具备良好开发技能、职业素养，并且能够兼通行业应用的移动互联应用开发工程师，帮助其实现产品快速迭代和上市，赢得竞争优势。

“联想移动互联网应用软件开发实战系列”是以真实行业产品研发项目任务为主线，采用 CDIO（构思-Conceive、设计-Design、实现-Implement 和运作-Operate）工程教育模式，辅以扩展知识（扩展-Extend），并结合企业行动式学习实践进行整体教学设计的实战教材。

CDIO 工程教育模式是以产品研发到产品运行的生命周期为载体，让参与者以主动的、实践的、课程之间有机联系的方式学习。CDIO 工程教育模式来源于工程实践，因此，CDIO 工程教育模式可以完美地应用于工程实践。在实战系列丛书中，我们对 CDIO 工程教育模式进行了扩充，加入了“扩展（Extend）”，从而把一个实战项目划分为五大任务：构思、设计、实现、运作和扩展，分别对应于移动互联应用软件项目开发中的五大阶段：需求分析、概要设计和详细设计、编码和测试、产品生命周期管理（软件维护和数据分析），以及贯穿始终的营销推广。对于五大任务，每个任务划分为若干个子任务；对于每个任务或子任务，同样应用 CDIOE 工程教育模式，其中，任务或子任务中的“扩展（Extend）”代表相关知识点的扩充和延展。这种双层嵌套的 CDIOE 工程教育模式，我们称之为“CDIOE 双循环工程教育模式”。

本丛书旨在为“软硬结合、懂行业、有素养”的高技能型人才培养提供实用、详细、结构化的实战项目训练支撑。

本丛书不重点讲解过多的方法论，而是把各种能力培养植入整个学习过程，通过每个任务的完成和总结，潜移默化地去启发读者，培养其从在校生向企业人转变过程中所需要的综合素质。除从事移动互联应用软件开发所必备的需求分析、产品设计、代码编写、代码测试、文档编写能力外，读者还可以通过实战中的角色扮演、过程参与、个人/集体分享等活动，在整个学习与实训过程中培养未来在企业中的基于个人绩效目标的管理能力、主动思考能力、团队沟通能力、企业内外部宣讲能力、时间管理，以及自我驱动的学习能力。

关于《Android 项目实战——智能农业移动管理系统开发》

《Android 项目实战——智能农业移动管理系统开发》是将智能农业典型需求和安卓编程相结合的移动互联应用软件开发实战项目。在本教材中，读者可以锻炼通用安

卓编程技巧，以及在智能农业典型场景下的行业应用编程能力。

请读者在本教材的学习过程中跟随项目导师，按照书中章节逐一实践，同时重视如下五个方面的体验与积累：

(1) 真实体验：在实战项目正式启动前，请在教师的指导下，对即将要开发产品的成品进行体验和使用，从用户角度去观察它的界面、功能、易用性等，并做好记录。

(2) 知识准备：在“真实体验”的基础上，对知识准备度进行自我差距分析，从“如果我来开发，还需要哪些知识”的假设出发，通过团队分享、与指导教师沟通来对前期所学知识进行回顾，对于已经确认掌握的知识可以稍作复习，对于部分掌握或仍未掌握的知识需要强化学习，并做好记录。

(3) 重现性实践：以 CDIO 为主线，按照本教材指导，逐一完成所规定的任务，以“真实体验”中的产品为标准，将产品重现。通过重现性实践来提升自身对所学理论知识的应用能力，同时获得每项任务中所提供的扩展能力。

(4) 反思与分享：“取他人之长，补己之所短”，通过不断的小组内分享以及小组间的分享，反思自己的不足，同时借鉴他人的经验，按照书中所提供的表格做好记录，并在此类过程中锻炼自己的主动思考和宣讲能力。

(5) 创新性发展：如果您有“真实体验”产品的改进建议，如：UI、代码专业性、规范性等，或教材本身优化或更新的相关建议，或全新的 Apps 开发成品，可以将信息反馈至 edusupport@lenovo.com，经联想专业团队评判并采纳后，联想将根据您的贡献，给予一定奖励。

以上这五个方面并非是需要读者按部就班进行的具体任务，而是在实训过程中需要读者不断通过具体任务来实践的学习指引。相信本教材会带领读者进入联想项目化实战的精彩世界。

本教材按照项目实战过程，划分为以下六章：

第 1 章实战导论：主要介绍实战项目中所必需的智能农业行业背景信息、知识准备度自检、实战分组和训练规则、学习目标设立等，属于前导课程。

第 2 章项目构思：涵盖产品开发在需求分析阶段的任务集，聚焦在从客户需求到软件功能需求翻译这一过程中的客户沟通、需求分析能力培养。

第 3 章项目设计：涵盖基于需求的概要设计及详细设计任务集，由两个子任务组成，聚焦软件设计能力培养。

第 4 章项目实现：涵盖具体的编码和测试任务集，也是本教材的重点，由 15 个子任务组成，聚焦代码阅读、编写和纠错能力培养。

第 5 章项目运作：涵盖产品开发完成后的签署与发布任务集，聚焦产品文档编写、产品功能宣讲能力培养。

第 6 章项目扩展：涵盖产品上市后，转向产品生命周期管理阶段任务集，聚焦软件维护领域所需具备的能力培养。

读者对象

有意从事移动互联应用软件开发/测试（安卓方向）的应届毕业生或在校生。

目 录

第 1 章 实战导论	1
1.1 智能农业概述	1
1.2 智能农业开发人才能力需求	1
1.3 目标产品真实体验	2
1.4 智能农业实战项目教学前导	6
1.5 智能农业实战项目教学目标	7
1.6 智能农业实战项目团队组建	9
1.7 智能农业实战项目考核方式	11
第 2 章 项目构思(Conceive)——需求分析	12
2.1 任务构思(Conceive)	12
2.2 任务设计(Design)	13
2.3 任务实现(Implement)	18
2.4 任务运作(Operate)	18
2.5 任务扩展(Extend)	23
第 3 章 项目设计(Design)——概要设计和详细设计	25
子任务 1 项目设计(Design)——概要设计	25
3.1.1 任务构思(Conceive)	25
3.1.2 任务设计(Design)	25
3.1.3 任务实现(Implement)	26
3.1.4 任务运作(Operate)	26
3.1.5 任务扩展(Extend)	28
子任务 2 项目设计(Design)——详细设计	28
3.2.1 任务构思(Conceive)	28
3.2.2 任务设计(Design)	28
3.2.3 任务实现(Implement)	29
3.2.4 任务运作(Operate)	29
3.2.5 任务扩展(Extend)	31
第 4 章 项目实现(Implement)——编码和测试	32
子任务 1 实现启动 App 后的 Splash 界面和用户引导界面	32
4.1.1 任务构思(Conceive)	32
4.1.2 任务设计(Design)	35
4.1.3 任务实现(Implement)	35

4.1.4	任务运作(Operate)	41
4.1.5	任务扩展(Extend)	42
子任务 2	实现主界面的布局	42
4.2.1	任务构思(Conceive)	42
4.2.2	任务设计(Design)	44
4.2.3	任务实现(Implement)	44
4.2.4	任务运作(Operate)	52
4.2.5	任务扩展(Extend)	53
子任务 3	实现客户端与服务端的网络连接	53
4.3.1	任务构思(Conceive)	53
4.3.2	任务设计(Design)	55
4.3.3	任务实现(Implement)	55
4.3.4	任务运作(Operate)	63
4.3.5	任务扩展(Extend)	64
子任务 4	实现用户的注册和登录	64
4.4.1	任务构思(Conceive)	64
4.4.2	任务设计(Design)	67
4.4.3	任务实现(Implement)	67
4.4.4	任务运作(Operate)	78
4.4.5	任务扩展(Extend)	79
子任务 5	实现客户端实时读取沙盘上传感器的数据	79
4.5.1	任务构思(Conceive)	80
4.5.2	任务设计(Design)	81
4.5.3	任务实现(Implement)	81
4.5.4	任务运作(Operate)	87
4.5.5	任务扩展(Extend)	88
子任务 6	在任务栏生成警报通知	88
4.6.1	任务构思(Conceive)	88
4.6.2	任务设计(Design)	90
4.6.3	任务实现(Implement)	91
4.6.4	任务运作(Operate)	96
4.6.5	任务扩展(Extend)	97
子任务 7	手动打开或关闭沙盘上的受控设备	98
4.7.1	任务构思(Conceive)	98
4.7.2	任务设计(Design)	101
4.7.3	任务实现(Implement)	101
4.7.4	任务运作(Operate)	106

目 录

4.7.5 任务扩展(Extend)	108
子任务 8 实现系统设置界面的布局	108
4.8.1 任务构思(Conceive)	108
4.8.2 任务设计(Design)	113
4.8.3 任务实现(Implement)	113
4.8.4 任务运作(Operate)	124
4.8.5 任务扩展(Extend)	125
子任务 9 实现应用程序的国际化	125
4.9.1 任务构思(Conceive)	125
4.9.2 任务设计(Design)	127
4.9.3 任务实现(Implement)	127
4.9.4 任务运作(Operate)	131
4.9.5 任务扩展(Extend)	132
子任务 10 自动打开或关闭沙盘上的受控设备	132
4.10.1 任务构思(Conceive)	132
4.10.2 任务设计(Design)	135
4.10.3 任务实现(Implement)	135
4.10.4 任务运作(Operate)	139
4.10.5 任务扩展(Extend)	140
子任务 11 使用第三方控件 AChartEngine 绘制图形	140
4.11.1 任务构思(Conceive)	140
4.11.2 任务设计(Design)	141
4.11.3 任务实现(Implement)	142
4.11.4 任务运作(Operate)	148
4.11.5 任务扩展(Extend)	149
子任务 12 查询传感器在某个时间段内的历史数据	149
4.12.1 任务构思(Conceive)	149
4.12.2 任务设计(Design)	158
4.12.3 任务实现(Implement)	158
4.12.4 任务运作(Operate)	162
4.12.5 任务扩展(Extend)	163
子任务 13 实现系统的权限管理	164
4.13.1 任务构思(Conceive)	164
4.13.2 任务设计(Design)	167
4.13.3 任务实现(Implement)	167
4.13.4 任务运作(Operate)	172
4.13.5 任务扩展(Extend)	173

子任务 14 在服务端记录系统日志	173
4.14.1 任务构思(Conceive)	173
4.14.2 任务设计(Design)	175
4.14.3 任务实现(Implement)	175
4.14.4 任务运作(Operate)	184
4.14.5 任务扩展(Extend)	185
子任务 15 使用 Monkey 对 App 进行压力测试	185
4.15.1 任务构思(Conceive)	185
4.15.2 任务设计(Design)	187
4.15.3 任务实现(Implement)	187
4.15.4 任务运作(Operate)	187
4.15.5 任务扩展(Extend)	188
第 5 章 项目运作(Operate)——签名和发布	189
5.1.1 任务构思(Conceive)	189
5.1.2 任务设计(Design)	189
5.1.3 任务实现(Implement)	189
5.1.4 任务运作(Operate)	190
5.1.5 任务扩展(Extend)	191
第 6 章 项目扩展(Extend)——软件维护	192
6.1.1 任务构思(Conceive)	192
6.1.2 任务设计(Design)	192
6.1.3 任务实现(Implement)	192
6.1.4 任务运作(Operate)	196
6.1.5 任务扩展(Extend)	197
附件 1 《智能农业实战项目需求说明书》模板	198
附件 2 《智能农业实战项目概要设计说明书》模板	202
附件 3 《智能农业实战项目详细设计说明书》模板	210

第1章 实战导论

1.1 智能农业概述

农业是人类赖以生存的基础，也是国民经济的基础。我国是一个农业大国，政府高度重视农业的发展。在传统的农业生产中，农民对农作物浇水、施肥和打药全凭经验和感觉。随着移动互联网和物联网的发展，智能农业应运而生。

所谓“智能农业”就是充分应用现代信息技术成果，集成应用计算机与网络技术、物联网技术、音视频技术、3S 技术、无线通信技术及专家智慧与知识，实现农业可视化远程诊断、远程控制、灾变预警等智能管理。“智能农业”是云计算、传感网、3S 等多种信息技术在农业中综合、全面的应用，实现更完备的信息化基础支撑、更透彻的农业信息感知、更集中的数据资源、更广泛的互联互通、更深入的智能控制、更贴心的公众服务。

“智能农业”能够有效改善农业生态环境，显著提高农业生产经营效率，彻底转变农业生产者、消费者观念和组织体系结构。

1.2 智能农业开发人才能力需求

实践证明，在智能农业相关解决方案的应用中，将移动互联的元素与智能农业场景相结合，能够更大地提升其监控、监测的效率和便利性。智能农业移动管理系统也将是智能农业整体方案中不可或缺的组成部分之一。而从事此类移动应用开发的人员不仅需要具备熟练的软件编程能力，还需要具备传感器、Zigbee 等物联网技术的应用能力。作为软件能力和硬件能力兼备的综合型开发人才，其具体的能力需求如下：

- 软件编程能力(共 5 项):
 - Java 编程能力:能够理解面向对象编程的特点，使用 Java 标准版中的普通特性和高级特性，编写出 PC 端的中小型应用程序。
 - Android 编程能力:能够使用 Android SDK 和 ADT 插件，在 Application Framework 的基础上，独立开发一个完整的 Android Application。
 - 数据库应用编程能力:在所开发的应用系统中使用数据库进行数据的存储和读取，并对数据库进行优化以提高操作的性能。
 - 网络协议应用编程能力:能够根据系统及用户的特点，选用满足要求的网络接口

协议，并根据协议的基本语法和给定的库或接口，实现网络协议的编程。

- 代码调测能力：能够对本人或他人编写的代码进行静态检查，断点调试、单元测试、性能测试和压力测试，以确保代码的质量稳定可靠。
- 物联网技术应用能力（共 2 项）：
 - 传感器：能够在软件系统中读取和解析传感器的数据。
 - Zigbee：能够了解 Zigbee 协议的特性、组网通信方式和性能。

1.3 目标产品真实体验

通过前两节的介绍，相信读者对于智能农业整体概念和从事移动管理系统开发的人员能力要求都有所了解，本小节除介绍的智能农业系统网络拓扑（如图 1-3-1 所示）外，也请读者在项目导师的指导下，对于在本实战项目中所要开发产品的成品进行体验，并按照本节所提供的表格进行记录和备注。

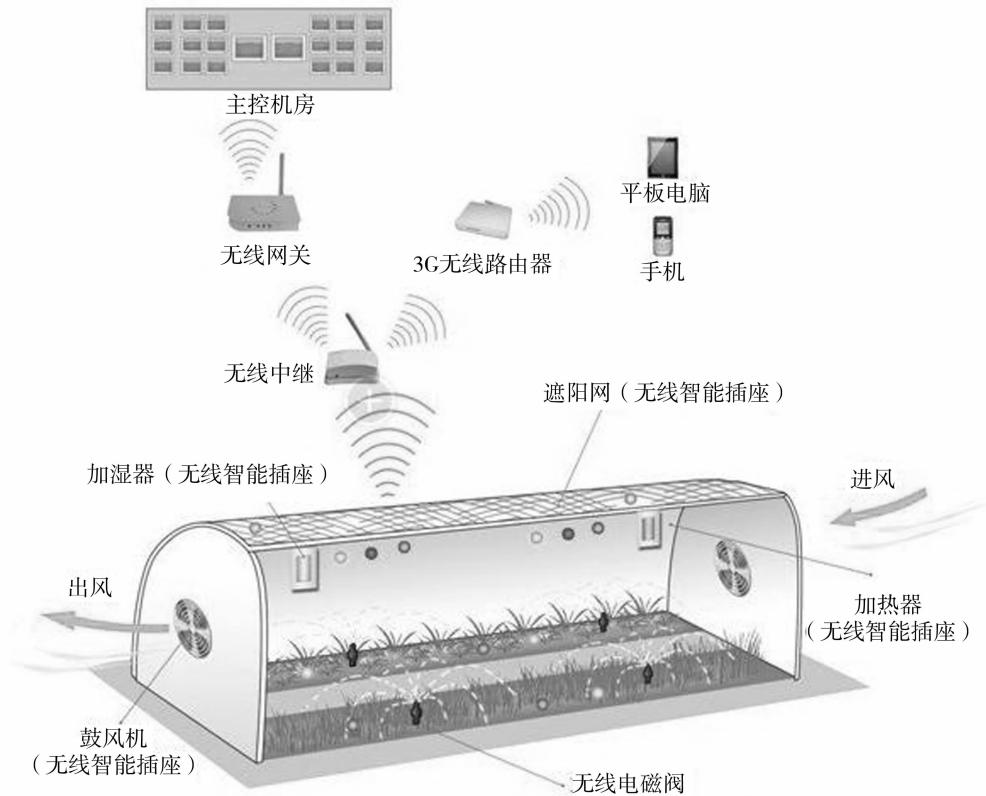


图 1-3-1 智能农业系统网络拓扑

本次实战项目是在图 1-3-1 基础上，进行开发的“联想智能农业管理系统”，其系统架构如图 1-3-2 所示。

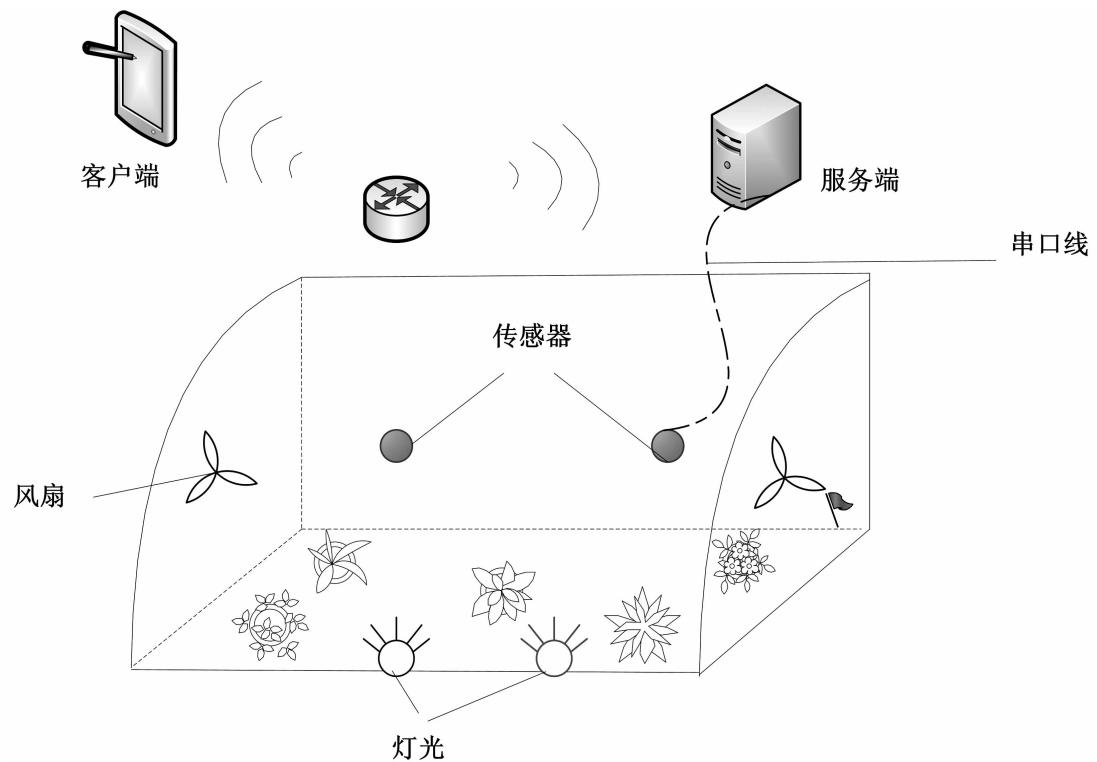


图 1-3-2 联想智能农业管理系统系统架构

联想智能农业管理系统的注册登录界面如图 1-3-3 所示。



图 1-3-3 联想智能农业管理系统的注册登录界面

用户点击“注册新用户”按钮，注册一个用户名和密码。然后在图 1-3-3 中输入注册的用户名和密码后，接着点击“登录”按钮，则进入环境指标界面，如图 1-3-4 所示。



图 1-3-4 环境指标界面

以上 Apps 的应用的数据均采自于联想智能农业仿真沙盘,如图 1-3-5 所示,当某个环境指标不在上述设置的范围之内时,通过软件控制,就会自动打开或关闭智能农业沙盘中的受控设备,例如:蜂鸣器、LED 灯和水泵等。

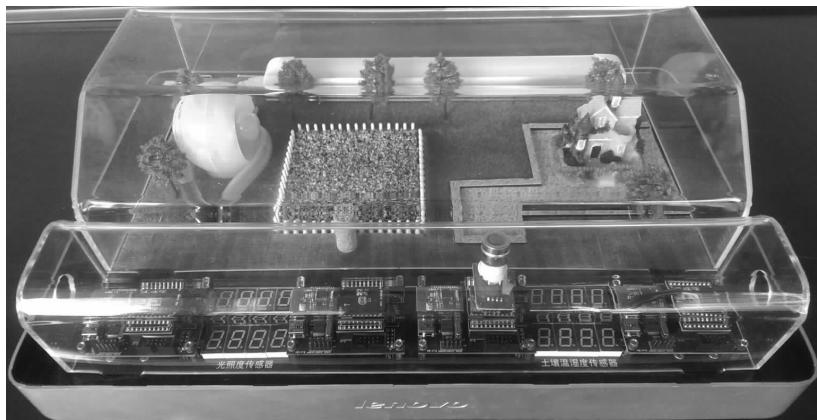


图 1-3-5 联想智能农业仿真沙盘

在对您即将要开发的产品直接体验后,让我们对承载本实战项目的操练环境做一个整体的介绍和回顾。本环境由三台设备组成:1台移动应用开发平台、1台智能农业数据网关和1个智能农业仿真沙盘,其中,移动应用开发平台用于模拟手机或平板电脑,智能农业数据网关用于模拟无线网关,智能农业仿真沙盘用于模拟农业大棚。三台设备的协同工作方式为:智能农业数据网关作为数据采集器,和智能农业仿真沙盘的单片机板通过串口相连,用于采集各种传感器的数据;同时,智能农业数据网关作为服务端,移动应用开发平台作为客户端,服务端和客户端的应用(App)进行通信,服务端将采集到的传感器数据实时发送到客户端的 App 进行显示,并根据设置的阀值打开和关闭智能农业仿真沙盘中的受控设备,包括:喷水器、LED 灯、风扇等。三台设备的拓扑连接如图 1-3-6 所示。

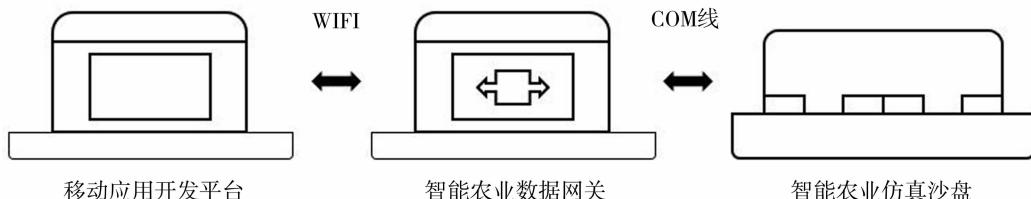


图 1-3-6 拓扑连接

移动应用开发平台如图 1-3-7 所示。



图 1-3-7 移动应用开发平台

智能农业数据网关如图 1-3-8 所示。

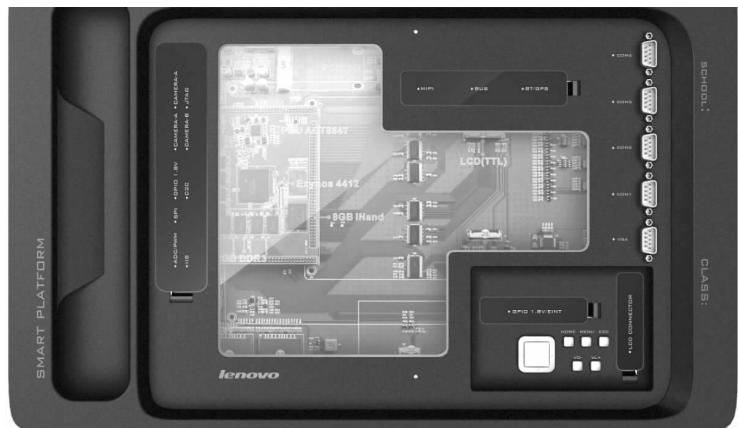


图 1-3-8 智能农业数据网关

反思与分享：

请利用 10~15 分钟, 回顾你本章节的收获并与周边同学分享, 同时将您在本章节中的疑问与答复记录在表 1-3-1 中。

表 1-3-1

疑问与答复记录表

序号	问题描述	答复	是否解决
1			<input type="checkbox"/> 是 <input type="checkbox"/> 否
2			<input type="checkbox"/> 是 <input type="checkbox"/> 否
3			<input type="checkbox"/> 是 <input type="checkbox"/> 否
4			<input type="checkbox"/> 是 <input type="checkbox"/> 否
5			<input type="checkbox"/> 是 <input type="checkbox"/> 否
6			<input type="checkbox"/> 是 <input type="checkbox"/> 否
7			<input type="checkbox"/> 是 <input type="checkbox"/> 否
8			<input type="checkbox"/> 是 <input type="checkbox"/> 否

1.4 智能农业实战项目教学前导

在正式启动开发项目前,结合您当前对本次实战项目的理解,让我们共同检查您对于本次实战项目所涉及知识的准备程度。请根据自身的实际情况,在表格 1-4-1 中进行勾选“掌握”、“部分掌握”及“未掌握”。为了保证您的实训效果,请对于已经“掌握”的知识可稍作回顾,而对于“部分掌握”和“未掌握”的知识,需要在课下进行加强和补充。

表 1-4-1

知识点准备度表

序号	主要知识点	准备度
1	移动互联应用软件开发的项目流程	<input type="checkbox"/> 掌握 <input type="checkbox"/> 部分掌握 <input type="checkbox"/> 未掌握
2	独立开发一个完整的 Android 应用程序,包括:需求分析、系统设计、编码、测试、签名和发布	<input type="checkbox"/> 掌握 <input type="checkbox"/> 部分掌握 <input type="checkbox"/> 未掌握
3	Java 和 Android 的编码规范	<input type="checkbox"/> 掌握 <input type="checkbox"/> 部分掌握 <input type="checkbox"/> 未掌握
4	应用程序启动后 Splash 效果的实现方式	<input type="checkbox"/> 掌握 <input type="checkbox"/> 部分掌握 <input type="checkbox"/> 未掌握
5	给定的图片资源,选择合适的布局方案和控件,实现界面的布局	<input type="checkbox"/> 掌握 <input type="checkbox"/> 部分掌握 <input type="checkbox"/> 未掌握
6	后台服务端和客户端之间的数据传输方式	<input type="checkbox"/> 掌握 <input type="checkbox"/> 部分掌握 <input type="checkbox"/> 未掌握
7	关系数据库的增删改查等常用操作	<input type="checkbox"/> 掌握 <input type="checkbox"/> 部分掌握 <input type="checkbox"/> 未掌握
8	日志文件在项目中的作用和用法	<input type="checkbox"/> 掌握 <input type="checkbox"/> 部分掌握 <input type="checkbox"/> 未掌握
9	多线程和 Handler 的常见用法	<input type="checkbox"/> 掌握 <input type="checkbox"/> 部分掌握 <input type="checkbox"/> 未掌握
10	Gesture 技术在触屏操作上的用法	<input type="checkbox"/> 掌握 <input type="checkbox"/> 部分掌握 <input type="checkbox"/> 未掌握
11	绘制图形相关的第三方库的用法	<input type="checkbox"/> 掌握 <input type="checkbox"/> 部分掌握 <input type="checkbox"/> 未掌握
12	常见模块(如登录注册模块和系统设置模块)的实现方式	<input type="checkbox"/> 掌握 <input type="checkbox"/> 部分掌握 <input type="checkbox"/> 未掌握
13	定时采集各种传感器数据的实现方式	<input type="checkbox"/> 掌握 <input type="checkbox"/> 部分掌握 <input type="checkbox"/> 未掌握

小贴士：

掌握：不仅概念清晰，而且会使用核心技术和常用技术。

部分掌握：概念清晰，但只会使用部分核心技术和常用技术。

未掌握：概念不清晰或无概念。

1.5 智能农业实战项目教学目标

通常情况下，编者会固定教材的教学目标，读者以此为目标进行学习。本书力求以读者视角进行教学目标的设定，每位读者可根据自身的实际情况，利用 SMART 原则及布鲁姆的“教学目标分类法”设定自己在本项目中的学习目标，并通过班级讨论的形式，最终形成全班的教学目标，并为之努力。具体设置步骤如下：

- (1) 在草稿纸上写出您当前所有的学习目标(注意 SMART 原则和布鲁姆方法)。
- (2) 邀请班级 50% 以上同学阐述自己的目标，以及设置目标的原因。
- (3) 最终确定全班在此项目中的学习目标，并填入表 1-5-1 中。

表 1-5-1

教学目标描述表

序号	教学目标描述
	例：两周后，在 Java 编程方面，能够描述面向对象编程的特点，使用 Java 标准版中的普通特性(如：数据类型、变量、操作符、流程控制、继承等)，编写出小型的基于 PC 端的应用程序
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	

概念澄清：

➤《SMART 原则》：是企业目标管理中的一种常用方法。目标管理的任务是有效地进行员工目标的制定和控制以达到更好的工作收益，由管理学大师彼得·杜拉克于 1954 年首先提出。SMART 原则中的“S”、“M”、“A”、“R”、“T”五个字母分别对应了五个含义：

- ◆ 目标必须是具体的(Specific)
- ◆ 目标必须是可以衡量的(Measurable)
- ◆ 目标必须是可以达到的(Attainable)
- ◆ 目标是要与其他目标具有一定的相关性(Relevant)
- ◆ 目标必须具有明确的截止期限(Time-bound)

无论是制定团队的工作目标还是员工的绩效目标都必须符合上述原则，五个原则缺一不可。

制定的过程也是自身能力不断增长的过程，目标明确会使我们的学习和工作更加有条理，利于在特定的时间段内高效取得预期的成果。

➤《布卢姆教育目标分类》：在教学目标设定过程中，主要可以利用其中在认知领域的教育目标的划分，请重点关注每类提示部分的动词，并将其使用在您的目标描述中。

◆ 知道(knowledge)

知道是指认识并记忆。这一层次所涉及的是具体知识或抽象知识的辨认，用一种非常接近于学生当初遇到的某种观念和现象时的形式，回想起这种观念或现象。

提示：回忆、记忆、识别、列表、定义、陈述、呈现。

◆ 领会(comprehension)

领会是指对事物的领会，但不要求深刻的领会，而是初步的，可能是肤浅的。其包括转化、解释、推断等。

提示：说明、识别、描述、解释、区别、重述、归纳、比较。

◆ 应用(application)

应用是指对所学习的概念、法则、原理的运用。它要求在没有说明问题解决模式的情况下，学会正确地把抽象概念运用于适当的情况。这里所说的应用是初步的直接应用，而不是全面地、通过分析、综合地运用知识。

提示：应用、论证、操作、实践、分类、举例说明、解决。

◆ 分析(analysis)

分析是指把材料分解成它的组成要素部分，从而使各概念间的相互关系更加明确，材料的组织结构更为清晰，详细地阐明基础理论和基本原理。

提示：分析、检查、实验、组织、对比、比较、辨别、区别。

◆ 综合(synthesis)

综合是以分析为基础，全面加工已分解的各要素，并再次把它们按要求重新地组合成整体，以便综合地创造性地解决问题。它涉及具有特色的表达，制定合理的计划和可实施的步骤，根据基本材料推出某种规律等活动。它强调特性与首创性，是高层次的要求。

提示：组成、建立、设计、开发、计划、支持、系统化。

◆评价(evaluation)

这是认知领域里教育目标的最高层次。这个层次的要求不是凭借直观的感受或观察的现象做出评判,而是理性地深刻地对事物本质的价值做出有说服力的判断,它综合内在与外在的资料、信息,做出符合客观事实的推断。

提示:评价、估计、评论、鉴定、辨明、辩护、证明、预测、预言、支持。

1.6 智能农业实战项目团队组建

1. 组建团队

当您完成以上内容后,让我们共同进入属于你的开发项目吧!请按照如下步骤组建5人开发团队:

(1)请班级成员按照如下三种学习风格进行站队,并领取该种风格的色卡,每人仅能选择一种风格:

①理论型:具备较好的理论功底,能够快速学习,掌握新知识。

②动手型:具备较好动手能力,对于实际操作能够快速上手。

③总结型:具备较好的问题解决能力,善于总结和表达,能够提出问题的本质。

(2)所有成员在色卡上写下姓名,并投入对应颜色的箱子中。

(3)由项目导师从3个箱子中,随机抽取5名成员,自动组成一个开发团队(若成员数量不足5的倍数,可以分散到其他各组中,团队成员数不超过7人)。

2. 团队项目角色划分

作为移动互联应用软件开发项目,通常整个开发团队主要有8个角色:项目经理、需求分析师、软件架构师、UI设计师、系统架构师、配置工程师、开发工程师、测试工程师,其关键职责如表1-6-1所示。在实际工作中,可以根据具体开发项目范围的大小,将角色进行合并或省略。

表1-6-1

角色与关键职责

角色	关键职责
项目经理	(1)整合管理 (2)范围管理 (3)时间管理 (4)成本管理 (5)质量管理 (6)人力资源管理 (7)沟通管理 (8)风险管理 (9)采购管理 (10)干系人管理

(续表)

角色	关键职责
软件架构师	(1)在技术上对所有重要事情做出决定 (2)需求分析阶段,负责理解和管理非功能性系统需求,比如软件的可维护性、性能、复用性、可靠性、有效性和可测试性等;审查客户和市场人员所提出的需求,确认开发团队所提出的设计;协助需求分析师完成《需求规格说明书》 (3)设计阶段,负责整个软件架构、关键构件和接口的设计,协助系统分析师完成《系统概要设计说明书》和《系统详细设计说明书》 (4)编码阶段,作为开发工程师的顾问,为其解决技术疑难问题 (5)测试阶段,协助测试工程师完成系统测试和集成测试
需求分析师	(1)对客户需求进行调研 (2)收集整理客户需求,编写《需求规格说明书》 (3)得到客户对《需求规格说明书》的认可与签字 (4)协助软件架构师、系统设计师对需求进行理解
UI 设计师	(1)设计原型图 (2)设计效果图,包括视觉设计和交互设计 (3)制作切图
系统设计师	(1)协助需求分析师进行需求调研 (2)编写《系统概要设计说明书》和《系统详细设计说明书》 (3)指导软件工程师按系统概要设计说明书》和《系统详细设计说明书》进行编码实现
配置工程师	(1)制定配置管理计划 (2)建立并维护配置管理库 (3)建立并发布基线 (4)物理审计 (5)跟踪并关闭变更申请 (6)报告配置状态
开发工程师	(1)协助系统设计师完成《系统详细设计说明书》 (2)根据《需求规格说明书》、《系统概要设计说明书》和《系统详细设计说明书》进行编码 (3)编写开发文档 (4)对代码进行单元测试 (5)协助测试工程师进行系统及模块的测试
测试工程师	(1)编写测试计划、规划详细的测试方案、编写测试用例 (2)根据测试计划搭建和维护测试环境 (3)执行测试工作,提交测试报告。包括编写用于测试的自动测试脚本,完整地记录测试结果,编写完整的测试报告等相关的技术文档 (4)对测试中发现的问题进行详细分析和准确定位,与开发人员讨论缺陷解决方案 (5)提出对产品的进一步改进的建议,并评估改进方案是否合理;对测试结果进行总结与统计分析,对测试进行跟踪,并提出反馈意见 (6)为业务部门提供相应技术支持,确保软件质量指标

在本实训项目中,具体角色划分规则如下:

(1)团队5人均有统一角色,即开发工程师。团队需重视“联想智能农业管理系统”,每人至少完成2个以上功能模块的代码编写。

(2)5名开发工程师中,其中1名兼任项目经理,1名兼任系统架构师,1名兼任测试工程师。其作用在于,在后续的实训过程中,对于每个任务或子任务,都有不同角色的人员参与其中,承担相应的职责,产生相应的输出并代表本团队进行团队间分享与总结。

1.7 智能农业实战项目考核方式

智能农业实战项目的考核成绩由平时成绩和答辩成绩两部分组成。其中,平时成绩占40%,依据考勤、上课表现、实验成果、开发成果等综合得出;答辩成绩占60%,答辩和评审的主要规则如下:

(1)以项目小组为单位,每个小组的项目经理通过PPT介绍,展示项目的总体情况,时间为20分钟。

(2)每个小组的成员通过PPT介绍自己在项目中的职责、收获以及可改进空间,时间为10分钟/人。

(3)每个小组成员的成绩=团队成绩×70%+个人成绩×30%。

第2章 项目构思(Conceive)——需求分析

- 本章目标：

- 领会需求分析在软件开发中的重要性
- 应用需求分析的常见方法
- 应用创建 WBS 的方法
- 应用 StarUML 工具的使用,能够绘制类图、时序图和流程图
- 应用 Balsamiq Mockups 工具的使用,能够绘制系统的原型图
- 应用 5W2H 分析法
- 领会计划会议、站立会议、评审会和复盘会的重要性
- 领会团队建设的塔克曼阶梯理论模型
- 应用《需求说明书》的编写要点

- 课时分配：

- 8 课时

2.1 任务构思(Conceive)

需求分析是指理解用户需求,就软件功能与客户达成一致,估计软件风险和评估项目代价,最终形成开发计划的一个复杂过程。在这个过程中,用户处在主导地位,需求分析工程师和项目经理要负责整理用户需求,为之后的软件设计打下基础。需求分析阶段结束后,要求得到相关的需求文档。

需求分析就是分析软件用户的需求是什么。如果投入大量的人力、物力、财力、时间,开发出的软件却没人要,那所有的投入都是徒劳。如果费了很大的精力,开发一个软件,最后却不满足用户的要求,从而要重新开发,这种返工是让人痛心疾首的。

需求分析之所以重要,就因为它具有决策性、方向性、策略性的作用,它在软件开发的过程中具有举足轻重的地位,大家一定要对需求分析具有足够的重视。在一个大型软件系统的开发中,它的作用要远远大于程序设计。

没有做好需求分析的后果如图 2-1-1 所示。

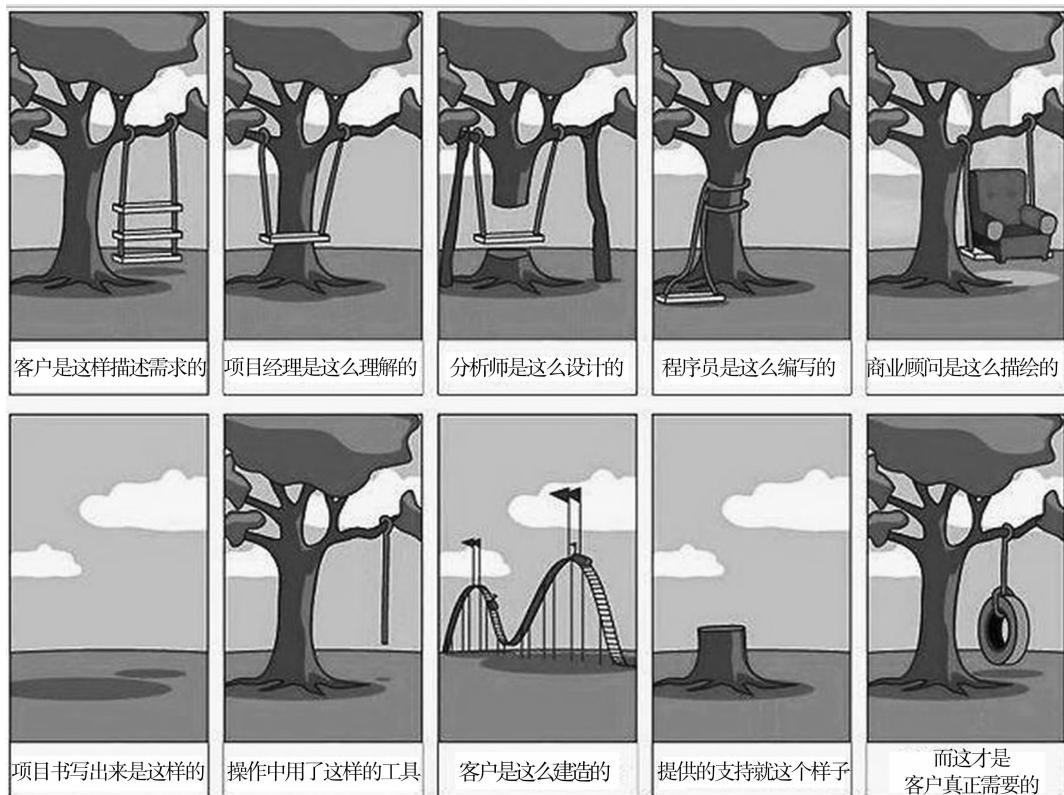


图 2-1-1 没有做好需求分析的后果

需求分析的任务就是解决“用户要做什么”的问题，就是要全面地理解用户的各项要求，并准确地表达所接受的用户需求，并且能够根据自己对用户需求的理解，劝说并诱导客户剔除不合理的要求。

2.2 任务设计 (Design)

智能农业实战项目的需求分析任务包含如下工作：

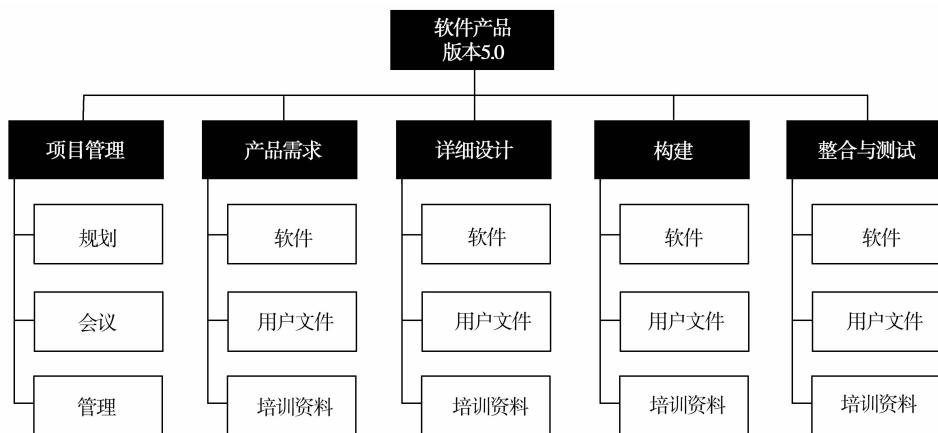
- (1) 结合对已有真实产品的体验，以及互联网中对智能农业行业现实应用的收集，列出产品的功能性需求。
- (2) 与潜在用户沟通并记录具体的需求。
- (3) 把实战项目的可交付成果和工作分解成较小的、更易于管理的组成部分，创建 WBS。
- (4) 使用建模工具 StarUML 画 UML 图，包括：用例图、类图和时序图。
- (5) 使用原型工具 Balsamiq Mockups 画原型图。
- (6) 编写《智能农业实战项目需求说明书》。

概念澄清：**WBS(Work Breakdown Structure, 工作分解结构)**

WBS 是项目管理重要的专业术语之一,是对项目团队为实现项目目标、创建可交付成果而需要实施的全部工作范围的层级分解。WBS 组织并定义了项目的总范围。分解是一种把项目范围和项目可交付成果逐步划分为更小、更便于管理的组成部分的技术。工作包是 WBS 最低层的工作,可对其成本和持续时间进行估算和管理。分解的程度取决于所需的控制程度,以实现对项目的高效管理。工作包的详细程度因项目规模和复杂程度而异。要把整个项目工作分解为工作包,通常需要开展以下活动:

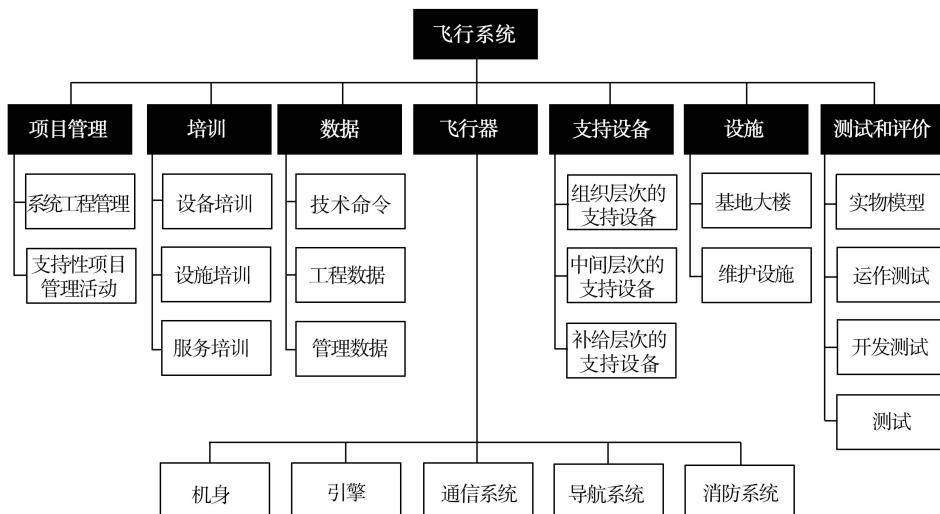
- 识别和分析可交付成果及相关工作;
- 确定 WBS 的结构和编排方法;
- 自上而下逐层细化分解;
- 为 WBS 组件制定和分配标识编码;
- 核实可交付成果分解的程度是否恰当。

对 WBS 上层的组件进行分解,就是要把每个可交付成果或组件的工作分解为最基本的元素,即可核实的产品、服务或成果。WBS 可以采用提纲式、组织结构图或能说明层级结构的其他形式。通过确认 WBS 下层组件是完成上层相应可交付成果的必要且充分的工作,来核实分解的正确性。不同的可交付成果可以分解到不同的层次。某些可交付成果只需分解到下一层,即可到达工作包的层次,而另一些则须分解更多层。工作分解得越细致,对工作的规划、管理和控制就越有力。但是,过细的分解会造成管理的无效耗费、资源使用效率低下、工作实施效率降低,同时造成 WBS 各层级的数据汇总困难。WBS 的示例如图 2-2-1 和图 2-2-2 所示。



这个WBS只是作为示例,不代表任何某个具体项目的完整项目范围,也不意味着此类项目仅此一种WBS分解方式。

图 2-2-1 WBS 示例 1



这个WBS只是作为示例，不代表任何某个具体项目的完整范围，也不意味着此类项目仅此一种WBS分解方式

图 2-2-2 WBS 示例 2

概念澄清：

UML(Unified Modeling Language,统一建模语言)

UML 是一个支持模型化和软件系统开发的图形化语言,为软件开发的所有阶段提供模型化和可视化支持,包括由需求分析到规格,到构造和配置。UML 从考虑系统的不同角度出发,定义了用例图、类图、对象图、状态图、活动图、序列图、协作图、构件图、部署图等 9 种图。这些图从不同的侧面对系统进行描述。系统模型将这些不同的侧面综合成一致的整体,便于系统的分析和构造。尽管 UML 和其他开发工具还会设计出许多派生的视图,但上述这些图和其他辅助性的文档是软件开发人员所见的最基本的构造。UML 统一了各种方法对不同类型的系统、不同开发阶段以及不同内部概念的不同观点,从而有效地消除了各种建模语言之间不必要的差异。它实际上是一种通用的建模语言,可以为许多面向对象建模方法的用户广泛使用。UML 建模能力比其他面向对象建模方法更强。它不仅适合于一般系统的开发,而且对并行、分布式系统的建模尤为适宜。UML 是一种建模语言,而不是一个开发过程。

工具介绍：

建模工具 StarUML

StarUML 是一款开放源码的 UML 开发工具,是由韩国公司主导开发出来的产品,可以直接到 StarUML 网站下载。StarUML 的主界面如图 2-2-3 所示。

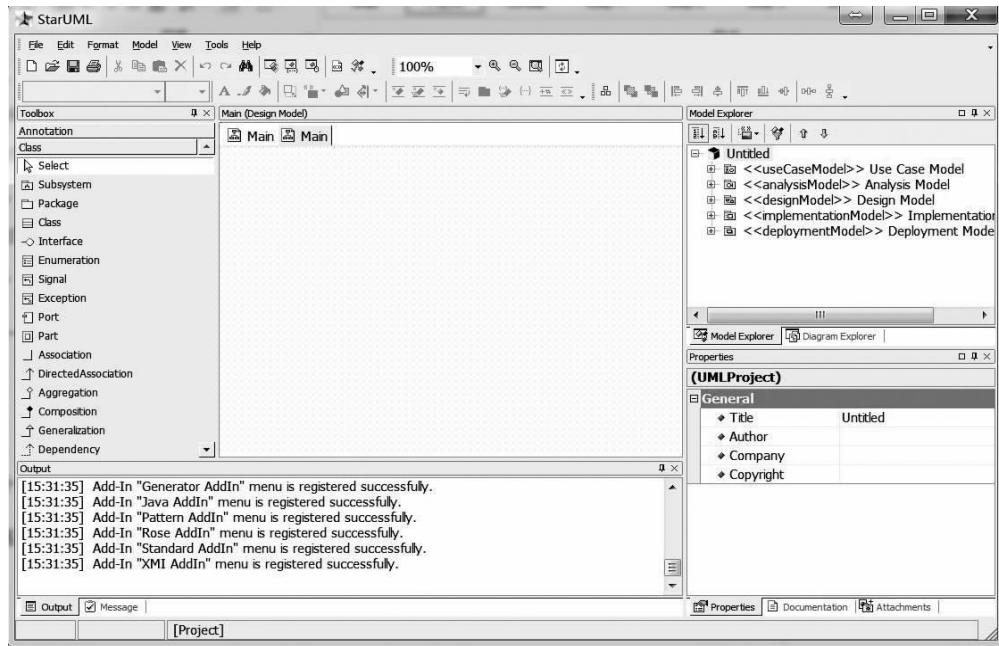


图 2-2-3 StarUML 主界面

使用 StarUML 绘制的系统用例图示例如图 2-2-4 所示。

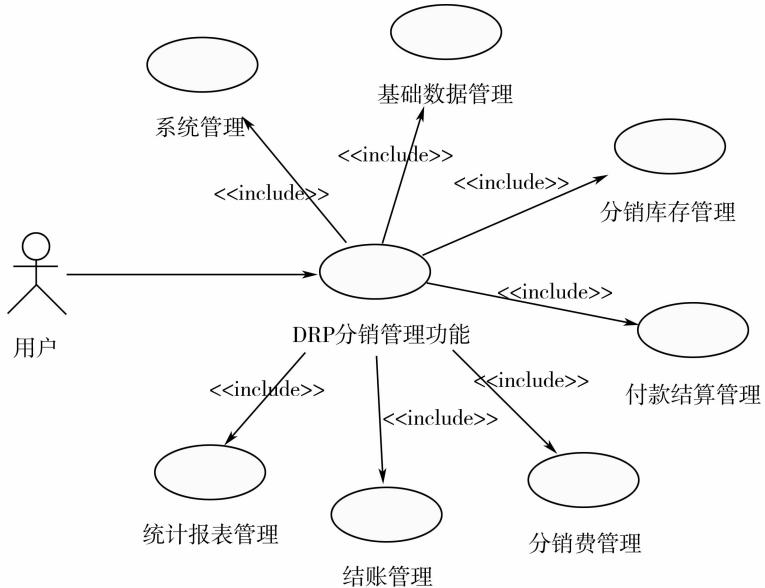


图 2-2-4 使用 StarUML 绘制的系统用例图示例

工具介绍：

原型工具 :Balsamiq Mockups

Balsamiq Mockups 是一种软件工程中快速原型的建立软件, 可以作为与用户交互的

一个界面草图，一旦客户认可可以作为美工开发 HTML 的原型使用。Balsamiq Mockups 的主界面如图 2-2-5 所示。

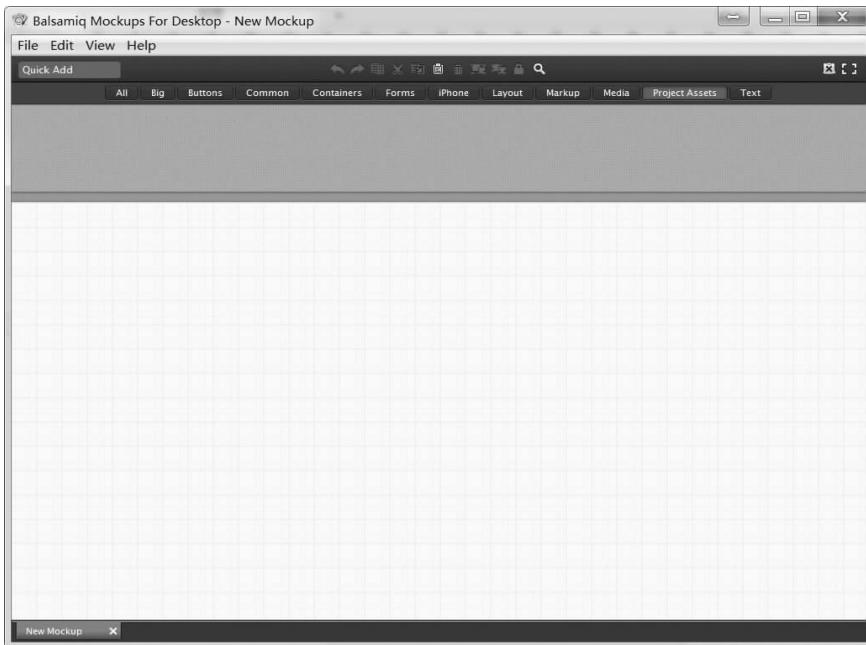


图 2-2-5 Balsamiq Mockups 主界面

使用 Balsamiq Mockups 绘制的系统原型图示例如图 2-2-6 所示。

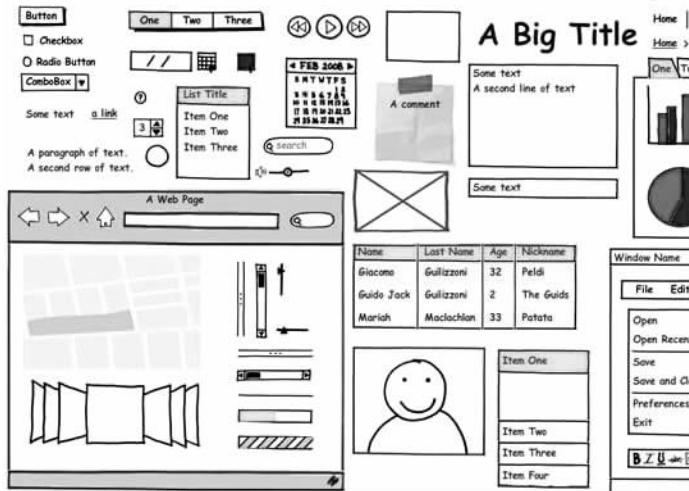


图 2-2-6 使用 Balsamiq Mockups 绘制的系统原型图示例

文档模板：

《智能农业实战项目需求说明书》模板

请见附件 1 所示。

2.3 任务实现(Implement)

智能农业实战项目的需求分析任务包含的角色如下：

- (1)项目经理(由一名开发工程师兼任);
- (2)软件架构师(由一名开发工程师兼任);
- (3)开发工程师;
- (4)测试工程师(由一名开发工程师兼任)。

角色扮演的过程如表 2-3-1 所示。

表 2-3-1 角色扮演过程

步骤	角色	工作	输出
1	项目经理 测试工程师	项目经理和测试工程师上网搜索智能农业行业的现实应用	智能农业行业的现实应用
2	项目经理 软件架构师 开发工程师 测试工程师	详细了解智能农业实战项目的仿真环境, 软件架构师和开发工程师扮演客户的角色, 与项目经理和测试工程师面对面地沟通, 询问并记录项目经理和测试工程师的需求	客户的真实需求
3	软件架构师 开发工程师	软件架构师和开发工程师创建 WBS	WBS
4	项目经理 测试工程师	项目经理和测试工程师使用工具 Balsamiq Mockups 画系统原型图	系统原型图
5	软件架构师 开发工程师	软件架构师和开发工程师使用工具 StarUML 画 UML 的用例图、类图和时序图	UML 用例图、类图和时序图
6	项目经理 软件架构师 开发工程师 测试工程师	所有团队成员共同编写《智能农业实战项目需求说明书》	《智能农业实战项目需求说明书》

2.4 任务运作(Operate)

在智能农业实战项目中, 任何任务开始之前, 都要召开计划会议(Planning Meeting)。在任务运作的过程中, 每天早上都要召开 15 分钟的站立会议(Standing Meeting)。

概念澄清：

计划会议 (Planning Meeting)

俗话说：“凡事预则立，不预则废”。在做任何事情之前，制定一个完善的计划都是非常必要的。在智能农业实战项目中，任何任务开始之前，所有团队成员都要参加计划会议，在计划会议上，采用 5W2H 分析法讨论如下七个问题：

- (1) WHAT——是什么？目的是什么？做什么工作？
- (2) HOW——怎么做？如何提高效率？如何实施？方法怎样？
- (3) WHY——为什么？为什么要这么做？理由何在？原因是什么？造成这样的结果为什么？
- (4) WHEN——何时？什么时间完成？什么时机最适宜？
- (5) WHERE——何处？在哪里做？从哪里入手？
- (6) WHO——谁？由谁来承担？谁来完成？谁负责？
- (7) HOW MUCH——多少？做到什么程度？数量如何？质量水平如何？费用产出如何？

讨论之后，将结论记录表格 2-4-1 中。

表 2-4-1

5W2H	结论
WHAT	
HOW	
WHY	
WHEN	
WHERE	
WHO	
HOW MUCH	

填完上面的表格之后，需求分析任务要做的具体工作及安排就很清楚了，此时，可以把这些任务再进一步细分，比如：收集需求、创建 WBS、绘制高层用例图、绘制某个模块的用例图、绘制类图、绘制时序图、绘制原型图等等，然后，把每个任务写在便签上，把便签张贴在如图 2-4-1 所示的看板上。

图 2-4-1 所示的看板分为三列：ToDo(要做的)、Doing(正在做的)和 Done(做完的)。计划会议结束之前，只需要把便签张贴在“ToDo”列，“Doing”和“Done”两列暂时没有便签。



图 2-4-1 看板 1

概念澄清：

站立会议 (Standing Meeting)

在任务运作的过程中，每天早上都要花 15 分钟的时间召开团队的站立会议，所有团队成员都要参加，并且必须站着开会。因此，站立会议也称“每日立会”。每日立会上每人都要汇报如下三个问题：

- (1) 我昨天做了什么；
- (2) 我今天要做什么；
- (3) 我遇到了什么困难。

每日立会的时间有限，必须控制在 15 分钟以内。因此，在每日立会上仅仅是抛出问题，不要讨论问题的解决方案。项目经理负责记录所有团队成员汇报的问题，并在会议结束之后逐一解决和落实。每人在汇报完上述三个问题之后，更新看板上的 ToDo、Doing 和 Done 三列，根据当前的状态把相关便签移动到对应的列，如图 2-4-2 所示。



图 2-4-2 看板 2

在《智能农业实战项目需求规格说明书》定稿之后,要召开需求分析任务的评审会(Review Meeting)和复盘会(Retrospective Meeting)。

概念澄清:

评审会(Review Meeting)

在每个任务完成之后,团队要召开评审会,所有团队成员都要参加。在评审会上,每个团队都向其他团队和指导老师展示当前任务的工作成果。其他团队成员和指导老师给出评价和反馈。以当前任务是否成功交付来评价任务的完成情况。评审会可分为两个阶段:

(1)建议收集:由项目经理作为代表,其余成员支撑,就本组的《智能农业实战项目需求规格说明书》进行讲解,其他组作为评审人员,对其输出进行评审,请将评审建议记录在表 2-4-2 中。

(2)建议回应:对其他组提的建议予以回应,并给出原因。

表 2-4-2

评审建议表

序号	建议描述	是否接纳	原因
1		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
2		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
3		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
4		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
5		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
6		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
7		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
8		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
9		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
10		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	

概念澄清:

复盘会(Retrospective Meeting)

在每个任务即将结束之前,团队要召开复盘会,所有团队成员都要参加。“复盘”是中国围棋的一个术语,蕴含着东方的智慧。2001 年,联想集团总裁柳传志第一次在联想提出“复盘”,2011 年将“复盘”作为联想的重要方法论向全球推广。复盘是联想文化的重要组成部分,复盘是实践联想之道的重要方法论,复盘是行动后的深刻反思和经验总结,复盘是一个不断学习、总结、反思、提炼和持续提高的过程。在联想,复盘是行动学习方法,更是一种习惯和一种文化,是提升组织智慧的手段。通过复盘,把失败转化为财富,把成功固化为能力。因此,在复盘会上,要总结哪些事情做得好,哪些事情做得不好,并且制定改进计划。联想的复盘模型如图 2-4-3 和图 2-4-4 所示。



图 2-4-3 联想复盘模型 1

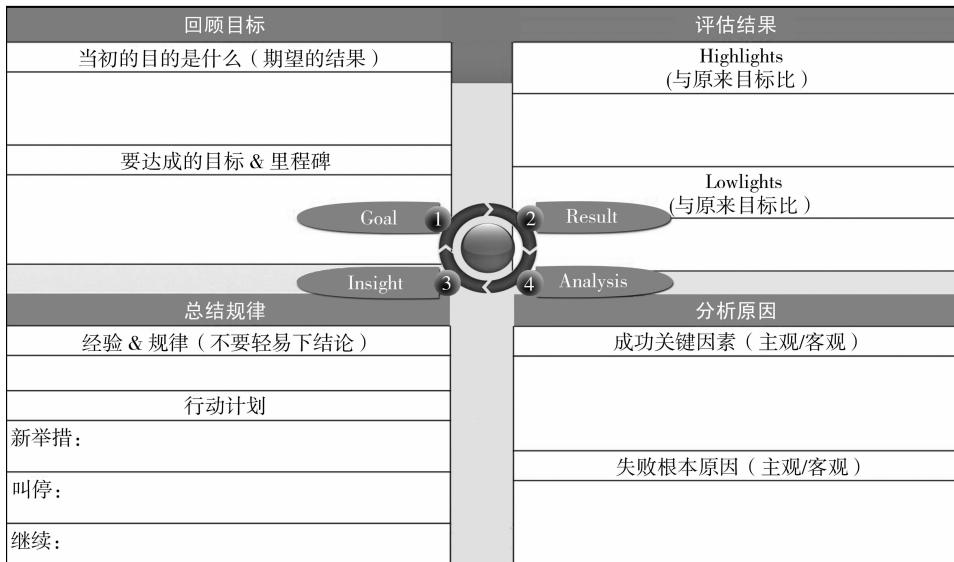


图 2-4-4 联想复盘模型 2

需求分析任务马上就要结束了,相信同学们在相互配合的过程中出现了很多问题,不过同学们不要着急,任何一个团队在组建初期都会遇到各种各样的问题,根据塔克曼阶梯理论,团队建设通常要经过如下5个阶段:

(1)形成阶段。在本阶段,团队成员相互认识,并了解项目情况及他们在项目中的正式角色与职责。团队成员倾向于相互独立,不一定开诚布公。

(2)震荡阶段。在本阶段,团队开始从事项目工作,制定技术决策和讨论项目管理办法。如果团队成员不能用合作和开放的态度对待不同观点和意见,团队环境可能变得事与愿违。震荡阶段的冲突最大,团队士气最低,也是克服冲突最难的阶段。

(3)规范阶段。在规范阶段,团队成员开始协同工作,并调整各自的工作习惯和行为来支持团队,团队成员开始相互信任。

(4)成熟阶段。进入这一阶段后,团队就像一个组织有序的单位那样工作。团队成员之间相互依靠,平稳高效地解决问题。

(5)解散阶段。在解散阶段,团队完成所有工作,团队成员离开项目。通常在项目可交付成果完成之后,再释放人员,解散团队;或者,在结束项目或阶段过程中解散团队。

同学们,让我们一起努力吧,在接下来的任务中,更加注重相互配合和协作,及时发

现并改正团队配合过程中出现的问题,早日让我们的团队进入成熟阶段!

2.5 任务扩展 (Extend)

了解原型工具 Axure RP 的用法。

此外,好的需求说明应达到如表 2-5-1 中所示的 10 个标准,请逐一核对《智能农业实战项目需求规格说明书》。

表 2-5-1

好的需求标准

序号	标准	含义
1	正确	需求规格说明书应当正确地反映用户的真实意图,“正确”是《需求规格说明书》最重要的属性。如果“不正确”仅仅是由于错别字造成的,那么多检查几遍文档就能解决问题。真正的困难是开发者和用户自己都不明白用户究竟“想要什么”和“不要什么”。为确保需求是正确的,开发方和用户必须对《需求规格说明书》进行确认
2	清楚	清楚的需求让人易读易懂。清楚的反义词是“难读”、“难理解”。你可以采用反问的方式来判断需求文档是否清楚:文档的结构、段落是否乱七八糟?上下文是否不连贯?文档的语句是否含糊其词、啰里啰嗦?看了半天是否还不明白需求究竟是什么
3	无二义性	“无二义性”是指每个需求只有唯一的含义。如果一个人说的话,不同的人可能有不同的理解,那么这句话就有二义性。如果需求存在二义性,将会导致人们误解需求而开发出偏离需求的产品。为了使需求无二义性,人们在写《需求规格说明书》时措词应当准确,切勿模棱两可
4	一致	“一致”(Consistent)是指《需求规格说明书》中各个需求之间不会发生矛盾。矛盾常常潜伏在需求文档的上下文中
5	必要	《需求规格说明书》中的各项需求对用户而言应当都是必要的。可以把“必要”比喻为“雪中送炭”。“必要”往前一步,要么是“画蛇添足”要么是“锦上添花”。“画蛇添足”显然是坏事,会导致开发人员多干一些吃力不讨好的工作。所以要尽量剔除需求规格说明书中“画蛇添足”的那些需求。“锦上添花”是好事,可能会让用户获得比期望更多的喜悦,但是眼前用户不会为此多付钱。开发者应当集中精力先完成必要的需求,如果条件允许则再做“锦上添花”的需求。为了避免主次颠倒,应当在《需求规格说明书》中将那些“锦上添花”的需求设置为较低的优先级
6	完备	“完备”(Complete)是指《需求规格说明书》中没有遗漏一些必要的需求。人们往往倾向于关注系统的特色功能,而忽视了其他一些不起眼的但却是必需的功能。不完备的《需求规格说明书》将导致产生功能不完整的软件,用户在使用该软件时可能无法完成预期的任务

(续表)

序号	标准	含义
7	可实现	《需求规格说明书》中的各项需求对开发方而言应当都是可实现的(Attainable)。“可实现”意味着在技术上是可行的，并且满足时间、费用、质量等约束。营销人员和用户谈生意时，为了能拿到“单子”，他们往往对用户提出的需求“来者不拒”。吹牛皮虽然不犯法，但是《需求规格说明书》可是白纸黑字。经过双方确认的《需求规格说明书》相当于商业合同，如果开发方不能够实现《需求规格说明书》中的内容，那就是违约，可能会被罚款。对于合同项目，如果开发方不能确信某些需求是否可实现，则应事先与用户协商，达成一致的处理意见，避免将来发生商业纠纷
8	可验证	《需求规格说明书》中的各项需求对用户方而言应当都是可验证的(Verifiable)。如果需求是不可验证的，那么用户就无法验收软件，可能会发生商业纠纷。例如，摩天大楼的一项需求是“抗十二级台风”，这个需求看起来堂而皇之，但是如何验证呢？当摩天大楼完工后验收时，用户又不是巫师，他怎能造个十二级台风来试验？如果双方都认可“采用计算机模拟十二级台风”等效于实际测试，那么这项需求就是“可验证”的
9	确定优先级	为什么要确定需求的“优先级”？理论上讲，软件的所有需求都应当被实现。但是在现实之中，项目存在“进度、费用、人力资源”等限制。在项目刚刚开始的时候，开发方和客户比较乐观，什么都要做，可是做着做着，人们常常会面临“进度延误、费用超支、人员不足”等问题，这时就乱套了。人们想出了“取舍”办法：先做优先级高的需求，后做优先级低的需求，这样可以将风险降到最低。需求的优先级其实就是需求“轻重缓急”的分级表述，例如划分为“高、中、低”三级。一般地，由用户和开发方共同确定需求的优先级
10	阐述“做什么”而不是“怎么做”	《需求规格说明书》的重点是阐述“做什么”，而不是阐述“怎么做”。“怎么做”是系统设计和实现阶段的事情。国内的很多软件公司里，开发人员常常身兼数职，可能把需求开发、系统设计、编程等工作从头做到尾。所以他们在调查、分析、定义需求时，自然会想到“怎么做”，这并没有什么过错。如果在调查、定义需求时想好了“怎么做”，应该记录下来。关键是不要将“怎么做”写到需求规格说明书里面，记录在其他文档里就行了

第3章 项目设计(Design) ——概要设计和详细设计

- 本章目标：

- 应用软件项目的概要设计
- 应用软件项目的详细设计

- 课时分配：

- 8 课时

子任务1 项目设计(Design)——概要设计

- 任务目标：

- 应用软件项目的概要设计

- 课时分配：

- 4 课时

3.1.1 任务构思(Conceive)

概要设计的主要任务是把需求分析得到的用例图转换为软件结构和数据结构。软件结构设计的具体任务是：将一个复杂系统按功能进行模块划分、建立模块的层次结构及调用关系、确定模块间的接口及人机界面等。数据结构设计包括数据特征的描述、确定数据的结构特性以及数据库的设计。

3.1.2 任务设计(Design)

按照构思阶段所获取的信息，编写概要设计的主要任务包括：

- (1) 设计软件结构。
- (2) 设计数据结构。
- (3) 编写《智能农业实战项目概要设计说明书》。

文档模板：

《智能农业实战项目概要设计说明书》模板
请见附件 2 所示。

3.1.3 任务实现(Implement)

智能农业实战项目的子任务包含的角色如下：

- (1) 系统架构师(由一名开发工程师兼任);
- (2) 开发工程师。

角色扮演的过程如表 3-1-1 所示。

表 3-1-1 角色扮演过程

步骤	角色	工作	输出
1	系统架构师 开发工程师	设计软件结构	软件结构
2	系统架构师 开发工程师	设计数据结构	数据结构
3	系统架构师 开发工程师	编写《智能农业实战项目概要设计说明书》	《智能农业实战项目概要设计说明书》
4	系统架构师 开发工程师	两两交叉设计评审	设计评审纪要

3.1.4 任务运作(Operate)

所有团队成员召开计划会议，并将会议纪要记录在表 3-1-2 中。

表 3-1-2 会议纪要记录表

5W2H	结论
WHAT	
HOW	
WHY	
WHEN	
WHERE	
WHO	
HOW MUCH	

在计划会议结束之前，所有团队成员把子任务进一步细分为更小的任务，把这些任务写在便签上，并把便签贴在看板上的“ToDo”列。

所有团队成员每天早上召开 15 分钟的站立会议，每人汇报三个问题，并更新看板上的“ToDo”、“Doing”和“Done”三列。

第③章 项目设计 (Design) —— 概要设计和详细设计

所有团队成员相互配合协作,共同进行软件设计、结构设计、编写《智能农业实战项目概要设计说明书》,并两两交叉评审。

所有团队成员召开评审会,并填写表 3-1-3。

表 3-1-3

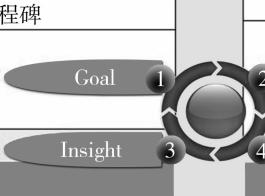
评审建议表

序号	建议描述	是否接纳	原因
1		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
2		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
3		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
4		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
5		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
6		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
7		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
8		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
9		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
10		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	

所有团队成员召开复盘会,并填写表 3-1-4。

表 3-1-4

复盘会总结表



回顾目标		评估结果	
当初的目的是什么(期望的结果)		Highlights (与原来目标比)	
要达成的目标&里程碑		Lowlights (与原来目标比)	
Goal	1	2	Result
Insight	3	4	Analysis
总结规律		分析原因	
经验&规律(不要轻易下结论)		成功关键因素(主观/客观)	
行动计划		失败根本原因(主观/客观)	
新举措:			
叫停:			
继续:			

同学们,根据塔克曼阶梯理论,你们的团队目前处于哪个阶段呢?如果还没有进入成熟阶段,那就让我们一起努力吧,在接下来的任务中,更加注重相互配合和协作,及时发现并改正团队配合过程中出现的问题,早日让我们的团队进入成熟阶段!

3.1.5 任务扩展(Extend)

课后练习：了解数据库建模工具 PowerDesigner 的使用。

子任务 2 项目设计(Design)——详细设计

- 任务目标：

- 应用软件项目的详细设计

- 课时分配：

- 4 课时

3.2.1 任务构思(Conceive)

详细设计是对概要设计的一个细化，就是详细设计每个模块的实现算法和所需的局部结构。详细设计必须遵循概要设计来进行。详细设计方案的更改，不得影响到概要设计方案；如果需要更改概要设计，必须经过项目经理的同意。详细设计，应该完成详细设计文档，主要是模块的详细设计方案说明。和概要设计一样，每个模块的详细设计文档都应该独立成册。

概要设计里面的数据库设计应该重点在描述数据关系上，说明数据的来龙去脉，在这里应该结合一下我们的结果数据，说明这些结果数据的源点，我们这样设计的目的和原因。详细设计里的数据库设计就应该是一份完善的数据结构文档，就是一个包括类型、命名、精度、字段说明、表说明等内容的数据字典。

概要设计里的功能应该是重点在功能描述，对需求的解释和整合，整体划分功能模块，并对各功能模块进行详细的图文描述，应该让读者大致了解系统做完后大体的结构和操作模式。详细设计则是重点在描述系统的实现方式，各模块详细说明实现功能所需的类及具体的方法函数，包括涉及到的 SQL 语句等。

3.2.2 任务设计(Design)

按照构思阶段所获取的信息，编写详细设计的主要任务包括：

- (1)为每个模块进行详细的算法设计。
- (2)为每个模块内的数据结构进行设计。
- (3)对数据库进行设计，即确定数据库的物理结构。
- (4)其他设计，比如代码设计、输入/输出格式设计和人机对话设计。
- (5)编写《智能农业实战项目详细设计说明书》。

文档模板：

《智能农业实战项目详细设计说明书》模板

请见附件 3 所示。

3.2.3 任务实现(Implement)

智能农业实战项目的该子任务包含的角色如下：

- (1) 系统架构师(由一名开发工程师兼任);
- (2) 开发工程师。

角色扮演的过程如表 3-2-2 所示。

表 3-2-2 角色扮演过程

步骤	角色	工作	输出
1	系统架构师 开发工程师	为每个模块进行详细的数据结构和算法设计	详细的数据结构和算法设计
2	系统架构师 开发工程师	对数据库进行详细设计	详细的数据库设计
3	系统架构师 开发工程师	编写《智能农业实战项目详细设计说明书》	《智能农业实战项目详细设计说明书》
4	系统架构师 开发工程师	两两交叉设计评审	设计评审纪要

3.2.4 任务运作(Operate)

所有团队成员召开计划会议，并将会议纪要记录在表 3-2-3 中。

表 3-2-3 会议纪要记录表

5W2H	结论
WHAT	
HOW	
WHY	
WHEN	
WHERE	
WHO	
HOW MUCH	

在计划会议结束之前，所有团队成员把子任务进一步细分为更小的任务，把这些任务写在便签上，并把便签贴在看板上的“ToDo”列。

所有团队成员每天早上召开 15 分钟的站立会议，每人汇报三个问题，并更新看板上

的“ToDo”、“Doing”和“Done”三列。

所有团队成员相互配合协作,共同进行数据结构设计、算法设计、数据库设计、编写《智能农业实战项目详细设计说明书》,并两两交叉评审。

所有团队成员召开评审会,并填写表 3-2-4。

表 3-2-4

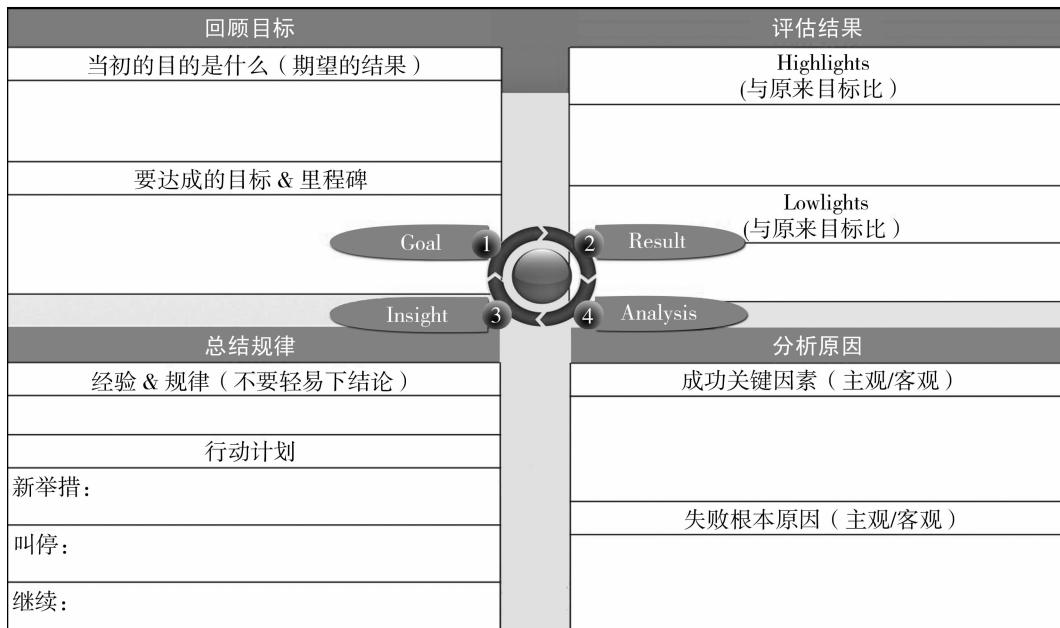
评审建议表

序号	建议描述	是否接纳	原因
1		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
2		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
3		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
4		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
5		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
6		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
7		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
8		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
9		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
10		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	

所有团队成员召开复盘会,并填写表 3-2-5。

表 3-2-5

复盘会总结表



同学们,根据塔克曼阶梯理论,你们的团队目前处于哪个阶段呢?如果还没有进入成熟阶段,那就让我们一起努力吧,在接下来的任务中,更加注重相互配合和协作,及时

发现并改正团队配合过程中出现的问题，早日让我们的团队进入成熟阶段！

3.2.5 任务扩展(Extend)

课后练习：强化数据结构、算法和数据库方面的知识和技能。

第 4 章 项目实现(Implement)

——编码和测试

- 本章目标：

- 应用 App 启动后 Splash 效果的实现方式
- 应用给定的图片资源,选择合适的布局方案和控件,实现界面的布局
- 应用后台服务端和客户端之间的数据传输方式
- 应用关系数据库的增删改查等常用操作
- 应用日志文件在项目中的作用和用法
- 应用多线程和 Handler 的常见用法
- 应用 Gesture 技术在触屏操作上的用法
- 应用绘制图形相关的第三方库的用法
- 应用常见模块(如登录注册模块和系统设置模块)的实现方式
- 应用定时采集各种传感器数据的实现方式

- 课时分配：

- 90 课时

子任务 1 实现启动 App 后的 Splash 界面和用户引导界面

- 任务目标：

- 实现启动 App 后的 Splash 界面和用户引导界面

- 课时分配：

- 8 课时

4.1.1 任务构思(Conceive)

很多商业 App 在启动时都有一个 Splash 启动界面,用来加载数据、展示 App 信息或植入广告。此外,当 App 安装后第一次启动时,很多商业 App 还有几个用户引导界面,

用来向用户介绍该 App 的主要功能特性或新版本中的新特性。用户可以通过手指向左滑屏来切换用户引导界面。当切换到最后一个用户引导界面时,通常再点击一个按钮即可进入到该 App 的主界面。

本任务对应的示例程序的运行效果如下:

(1)启动示例程序,进入 Splash 界面,如图 4-1-1 所示。



图 4-1-1 Splash 欢迎界面

(2)在 Splash 界面,程序会停留 3 秒钟,然后由 Splash 界面跳转到用户引导界面。在用户引导界面,通过手指向左滑屏依次显示三个智能农业系统的功能介绍图片,如图 4-1-2、图 4-1-3 和图 4-1-4 所示。



图 4-1-2 功能介绍 1

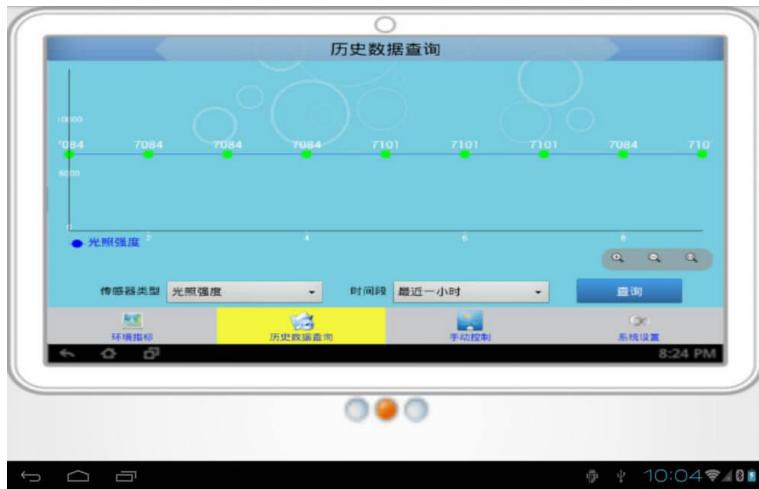


图 4-1-3 功能介绍 2



图 4-1-4 功能介绍 3

(3) 点击上图中的按钮【点击进入轻松体验】进入主界面,如图 4-1-5 所示。



图 4-1-5 进入主界面

(4) 再次启动示例程序, 进入 Splash 页面, 如图 4-1-1 所示。

(5) 在 Splash 界面停留 3 秒钟后直接跳转到主界面(没有跳转到用户引导界面), 如图 4-1-5 所示。

4.1.2 任务设计(Design)

➤ Splash 界面的任务设计

Splash 界面的常用实现方式有: 线程、Handler 和 Timer 等。本任务对应示例程序的主要设计思路是:

(1) 通过取出 SharedPreferences 中保存的启动值来判断本任务对应示例程序是否是第一次启动。

(2) 利用 Handler 发送一个延时消息。

(3) 根据消息中的 what 值进行界面的跳转。

➤ 用户引导界面的任务设计

用户引导界面的常用实现方式有: ViewPager 和 ViewFlipper。本任务对应示例程序的主要设计思路是:

(1) 使用 findViewById 获取布局文件中定义的 ViewPager 对象。

(2) 自定义 ViewPager 的适配器。

(3) 为 ViewPager 添加适配器和监听事件。

4.1.3 任务实现(Implement)

所有团队成员召开计划会议, 并将会议纪要记录在表 4-1-1 中。

表 4-1-1

会议纪要记录表

5W2H	结论
WHAT	
HOW	
WHY	
WHEN	
WHERE	
WHO	
HOW MUCH	

在计划会议结束之前, 所有团队成员把子任务进一步细分为更小的任务, 把这些任务写在便签上, 并把便签贴在看板上的“ToDo”列。

所有团队成员每天早上召开 15 分钟的站立会议, 每人汇报三个问题, 并更新看板上的“ToDo”、“Doing”和“Done”三列。

本任务对应示例程序的核心代码如下:

(1) lenovo.agriculture.activity.SplashActivity 类中的核心代码

/ * *

* 判断程序是否是第一次运行

* @return 是否第一次运行

```
* /
private boolean isFirstRun() {
    return mPrefrence.getBoolean("isFirst", true);
}

/* *
 * 控制跳转
 */
private void into() {
    if (isFirstRun()) {
        mHandler.sendEmptyMessageDelayed(ENTER_SPLASH, waitTime);
    } else {
        mHandler.sendEmptyMessageDelayed(ENTER_HOME, waitTime);
    }
}

/* *
 * 消息处理
*/
private void handleMessage() {
    mHandler = new Handler(new Handler.Callback() {
        @Override
        public boolean handleMessage(Message msg) {
            if (msg.what == ENTER_HOME) {
                // 进入主页面
                Intent mHomeIntent = new Intent(SplashActivity.this,
                    MainActivity.class);
                startActivity(mHomeIntent);
            } else {
                // 将 App 启动的值设为 false
                Editor mEditor = mPrefrence.edit();
                mEditor.putBoolean("isFirst", false);
                mEditor.commit();
                // 进入用户引导界面
                Intent mSplshIntent = new Intent(SplashActivity.this,
                    GuideActivity.class);
                startActivity(mSplshIntent);
            }
            finish();
            returnfalse;
        }
    });
}
```

(2) lenovo.agriculture.activity.GuideActivity 类中的核心代码

```

/* *
 * 初始化控件
 */
private void initView() {
    mPointSetLayout = (LinearLayout) findViewById(R.id.point_show_layout);
    mViewPager = (ViewPager) findViewById(R.id.picture_show_vp);
}

/* *
 * 为 ViewPager 设置相应的参数
 */
private void setViewPager() {
    mAdapter = new ViewPagerAdapter(this, mList);
    mViewPager.setAdapter(mAdapter);
    mViewPager.setOnPageChangeListener(this);
    mViewPager.setCurrentItem(0);
}

```

(3) lenovo.agriculture.adaper.ViewPagerAdapter 类中的核心代码

```

/* *
 * 自定义视图
 * @see android.support.v4.view.PagerAdapter#instantiateItem(android.view.ViewGroup,
 *      int)
 */
@Override
public Object instantiateItem(ViewGroup container, int position) {
    View view = (View) mInflater.inflate(R.layout.guide_item, container,
        false);
    RelativeLayout mLayout = (RelativeLayout) view
        .findViewById(R.id.picture);
    TextView mTextView = (TextView) view.findViewById(R.id.enter_tv);
    mLayout.setBackgroundResource(mList.get(position));
    mTextView.setBackgroundResource(R.drawable.control_bg);
    // 设置跳转按钮
    if (position == 2) {
        mTextView.setVisibility(View.VISIBLE);
        mTextView.setBackgroundColor(Color.GREEN);
    } else {
        mTextView.setVisibility(View.GONE);
    }
    // 跳转到主界面
    mTextView.setOnClickListener(new OnClickListener() {

```

```
@Override
public void onClick(View v) {
    GuideActivity mActivity = (GuideActivity) mContext;
    mContext.startActivity(new Intent(mContext, MainActivity.class));
    mActivity.finish();
}
});
((ViewPager) container).addView(view, 0);
return view;
}

(4)res/layout/splash.xml 中的核心代码
<? xml version="1.0" encoding="utf-8"? >
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/app_bg">

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:layout_margin="10dp"
        android:gravity="center"
        android:text="@string/welcome_prompt"
        android:textSize="75sp"/>

    <TextView
        android:id="@+id/software_version_tv"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/textView1"
        android:layout_centerHorizontal="true"
        android:textSize="75sp"/>

</RelativeLayout>
(5)res/layout/guide_show.xml 中的核心代码
<? xml version="1.0" encoding="utf-8"? >
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
```

```
    android:layout_height= "match_parent"
    android:orientation= "vertical">

    <LinearLayout
        android:layout_width= "match_parent"
        android:layout_height= "match_parent"
        android:layout_weight= "0.15"
        android:orientation= "vertical">

        <android.support.v4.view.ViewPager
            android:id= "@+id/picture_show_vp"
            android:layout_width= "match_parent"
            android:layout_height= "match_parent"/>
    </LinearLayout>

    <LinearLayout
        android:id= "@+id/point_show_layout"
        android:layout_width= "match_parent"
        android:layout_height= "match_parent"
        android:layout_weight= "0.89"
        android:gravity= "center_horizontal/top"
        android:orientation= "horizontal">
    </LinearLayout>

</LinearLayout>

(6)res/layout/guide_item.xml 中的核心代码
<? xml version= "1.0" encoding= "utf-8"? >
<RelativeLayout xmlns:android= "http://schemas.android.com/apk/res/android"
    android:id= "@+id/picture"
    android:layout_width= "match_parent"
    android:layout_height= "match_parent"
    android:orientation= "vertical">

    <TextView
        android:id= "@+id/enter_tv"
        android:layout_width= "wrap_content"
        android:layout_height= "wrap_content"
        android:layout_alignParentBottom= "true"
        android:layout_centerHorizontal= "true"
        android:text= "@string/enter_prompt"
        android:textSize= "25sp"/>

</RelativeLayout>
```

(7) res/values/strings.xml 中的核心代码

```
<? xml version= "1.0" encoding= "utf-8"? >
<resources>

    <string name= "app_name">Splash+Guide</string>
    <string name= "enter_prompt">点击进入轻松体验</string>
    <string name= "envir_title">环境指标界面</string>
    <string name= "history_title">历史查询界面</string>
    <string name= "control_title">手动控制界面</string>
    <string name= "welcome_prompt">欢迎进入 \n 智能农业系统! </string>
    <string name= "software_version">版本号:</string>

</resources>
```

(8) AndroidManifest.xml 中的核心代码

```
<? xml version= "1.0" encoding= "utf-8"? >
<manifest xmlns:android= "http://schemas.android.com/apk/res/android"
    package= "lenovo.agriculture.activity"
    android:versionCode= "1"
    android:versionName= "1.0" >

    <uses-sdk
        android:minSdkVersion= "15"
        android:targetSdkVersion= "21"/>
    <application
        android:allowBackup= "true"
        android:icon= "@drawable/ic_launcher"
        android:label= "@string/app_name"
        android:theme= "@style/AppTheme">
        <activity
            android:name= ".SplashActivity"
            android:label= "@string/app_name"
            android:screenOrientation= "landscape">
            <intent-filter>
                <action android:name= "android.intent.action.MAIN"/>
                <category android:name= "android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
        <activity android:name= ". BaseActivity"/>
        <activity
            android:name= ".MainActivity"
            android:screenOrientation= "landscape"/>
        <activity>
```

```

    android:name=".GuideActivity"
    android:screenOrientation="landscape"/>
</application>

</manifest>
```

4.1.4 任务运作(Operate)

开发工程师之间两两交叉代码评审，并根据评审纪要重构代码。

所有团队成员召开评审会，并填写表 4-1-2。

表 4-1-2 评审建议表

序号	建议描述	是否接纳	原因
1		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
2		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
3		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
4		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
5		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
6		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
7		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
8		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
9		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
10		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	

所有团队成员召开复盘会，并填写表 4-1-3。

表 4-1-3 复盘会总结表

回顾目标		评估结果	
当初的目的是什么（期望的结果）		Highlights (与原来目标比)	
要达成的目标 & 里程碑		Lowlights (与原来目标比)	
Goal	1	Result	2
总结规律		分析原因	
经验 & 规律（不要轻易下结论）		成功关键因素（主观/客观）	
行动计划		失败根本原因（主观/客观）	
新举措：			
叫停：			
继续：			

同学们,根据塔克曼阶梯理论,你们的团队目前处于哪个阶段呢?如果还没有进入成熟阶段,那就让我们一起努力吧,在接下来的任务中,更加注重相互配合和协作,及时发现并改正团队配合过程中出现的问题,早日让我们的团队进入成熟阶段!

4.1.5 任务扩展(Extend)

课后练习:在该示例项目的某个用户引导界面上添加音乐播放效果。

子任务 2 实现主界面的布局

- 任务目标:

- 实现主界面的布局

- 课时分配:

- 8 课时

4.2.1 任务构思(Conceive)

联想智能农业系统包括环境指标、历史查询、手动控制和系统设置四大功能模块,为了在系统主界面上展示这四大功能模块,需要对系统主界面的布局进行设计。

本任务对应的示例程序的运行效果如下:

(1)启动示例程序,切换到主界面,如图 4-2-1 所示。



图 4-2-1 程序主界面

(2)点击选项卡【历史查询】,切换到历史查询界面,如图 4-2-2 所示。

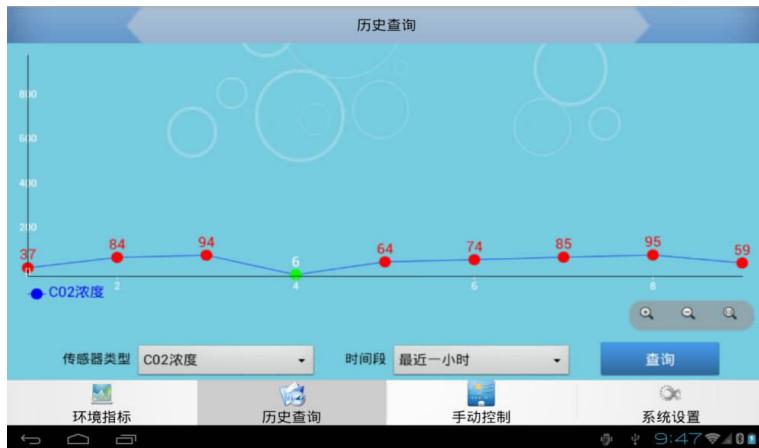


图 4-2-2 历史查询界面

(3) 点击选项卡【手动控制】，切换到手动控制界面，如图 4-2-3 所示。



图 4-2-3 手动控制界面

(4) 点击选项卡【系统设置】，切换到系统设置界面，如图 4-2-4 所示。



图 4-2-4 系统设置界面

4.2.2 任务设计(Design)

本任务对应示例程序的实现方式通常是 TabHost 或 TabHostFragment。当用户点击界面下方的某个 Tab 时，就会在界面的上方区域显示该 Tab 的具体内容。

4.2.3 任务实现(Implement)

所有团队成员召开计划会议，并将会议纪要记录在表 4-2-1 中。

表 4-2-1

会议纪要记录表

5W2H	结论
WHAT	
HOW	
WHY	
WHEN	
WHERE	
WHO	
HOW MUCH	

在计划会议结束之前，所有团队成员把子任务进一步细分为更小的任务，把这些任务写在便签上，并把便签贴在看板上的“ToDo”列。

所有团队成员每天早上召开 15 分钟的站立会议，每人汇报三个问题，并更新看板上的“ToDo”、“Doing”和“Done”三列。

本任务对应示例程序的核心代码如下：

(1) lenovo.agriculture.mainlayout.activity.MainActivity 类中的核心代码

```
/*
 * 初始化控件
 */
private void initView() {
    mTabHost = (FragmentTabHost) findViewById(android.R.id.tabhost);
    mTabHost.setup(this, getSupportFragmentManager(), R.id.realtabcontent);

    mTitleTV = (TextView) findViewById(R.id.title_tv);
    mTitleTV.setText(mTagStrings[0]);
}

/*
 * 添加视图和指示器
 */
private void addTabs() {
    for (int i = 0; i < mTagStrings.length; i++) {
```

```

TabSpec tag = mTabHost.newTabSpec(mTagStrings[i]).setIndicator(
        getTabItemView(i));
mTabHost.addTab(tag, fragments[i], null);
mTabHost.getTabWidget().getChildAt(i)
        .setBackgroundResource(R.drawable.bg);
}
}

```

(2) lenovo.agriculture.mainlayout.fragment.EnvirFragment 类中的核心代码

```

/*
 * 初始化数据
 */
private void initData() {
    mList = new ArrayList<SensorBean>();
    SensorBean mBean;
    String[] mStrings = getResources().getStringArray(R.array.sensor);
    for (int i = 0; i < mStrings.length; i++) {
        mBean = new SensorBean(mStrings[i]);
        mBean.setValue(3 * i);
        mBean.setMaxValue(2 * i + 3);
        mBean.setMinValue(2 * i);
        mList.add(mBean);
    }
}
/*
 * GridView 设置适配器
 */
private void setGridView() {
    mAdapter = new SensorGridAdapter(getActivity(), mList);
    gridView.setAdapter(mAdapter);
}

```

(3) lenovo.agriculture.mainlayout.adpater.SensorGridAdapter 类中的核心代码

```

/*
 * 自定义视图
 * @see android.widget.ArrayAdapter#getView(int, android.view.View,
 *      android.view.ViewGroup)
 */
public View getView(int position, View convertView, ViewGroup parent) {
    try {
        SensorBean itemInfo = (SensorBean) getItem(position);
        ViewHolder holder = null;
        if (convertView == null) {

```

```
convertView = mInflater.inflate(R.layout.sensor_grid_item ,  
        parent, false );  
holder = new ViewHolder();  
// 界面初始化  
holder.mBgLayout = (LinearLayout) convertView  
        .findViewById(R.id.sensor_item_layout );  
holder mNameTV = (TextView) convertView  
        .findViewById(R.id.sensor_name_text );  
holder.mStautsTV = (TextView) convertView  
        .findViewById(R.id.status_text );  
holder.mSetValueTV = (TextView) convertView  
        .findViewById(R.id.set_value_text );  
holder.mValueTV = (TextView) convertView  
        .findViewById(R.id.sensor_value_text );  
convertView.setTag(holder);  
} else {  
    holder = (ViewHolder) convertView.getTag();  
}  
  
if (itemInfo != null ) {  
    holder mNameTV.setText(itemInfo.getName());  
    holder.mValueTV.setText(" " + itemInfo.getValue());  
  
    // 显示设定值内容  
    String setStr = mContext.getString(R.string.set_value );  
    final int invalidMin = Integer.MIN_VALUE ;  
    int minV = itemInfo.getMinValue();  
    int maxV = itemInfo.getMaxValue();  
    if (minV > invalidMin && maxV > invalidMin) {  
        setStr += minV + " ~ " + maxV;  
    } else if (minV > invalidMin) {  
        setStr += ">" + minV;  
    } else if (maxV > invalidMin) {  
        setStr += "<" + maxV;  
    }  
    holder.mSetValueTV.setText(setStr);  
  
    // 根据传感器的值来判断当前是否应该告警  
    boolean isWarning = false ;  
    int value = itemInfo.getValue();
```

```

if (value > invalidMin
    && ((minV > invalidMin || maxV > invalidMin))) {
    if (minV > invalidMin && value < minV) {
        isWarning = true ;
    }
    if (maxV > invalidMin && value > maxV) {
        isWarning = true ;
    }
}

if (isWarning) {
    // 如果需要告警,背景则显示红色
    holder.mBgLayout.setBackgroundResource(R.color.card_bg_red);
    holder.mStautsTV.setText(R.string.warning);
} else {
    // 不必告警则显示绿色
    holder.mBgLayout
        .setBackgroundResource(R.color.card_bg_green);
    holder.mStautsTV.setText(R.string.normal);
}
}

} catch (Exception e) {
    e.printStackTrace();
}

return convertView;
}

```

(4)res/layout/activity_main.xml 中的核心代码

```

<? xml version="1.0" encoding="utf-8"? >
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <include
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        layout="@layout/title"/>

    <FrameLayout
        android:id="@+id/realtabcontent"

```

```
    android:layout_width= "match_parent"
    android:layout_height= "0dp"
    android:layout_weight= "9"
    android:background= "@drawable/app_bg"/>

<android.support.v4.app.FragmentTabHost
    android:id= "@+id/tabhost"
    android:layout_width= "match_parent"
    android:layout_height= "0dp"
    android:layout_weight= "1.2">
</android.support.v4.app.FragmentTabHost>

</LinearLayout>
(5)res/layout/envir_fragment.xml 中的核心代码
<? xml version= "1.0" encoding= "utf-8"? >
<GridView xmlns:android= "http://schemas.android.com/apk/res/android"
    android:id= "@+id/envir_gridview"
    android:layout_width= "match_parent"
    android:layout_height= "match_parent"
    android:horizontalSpacing= "20dp"
    android:numColumns= "3"
    android:padding= "5dp"
    android:verticalSpacing= "10dp">

</GridView>
(6)res/layout/sensor_grid_item.xml 中的核心代码
<? xml version= "1.0" encoding= "utf-8"? >
<LinearLayout xmlns:android= "http://schemas.android.com/apk/res/android"
    android:layout_width= "match_parent"
    android:layout_height= "wrap_content"
    android:background= "@color/card_bg_green"
    android:orientation= "vertical">

<LinearLayout
    android:id= "@+id/sensor_item_layout"
    android:layout_width= "match_parent"
    android:layout_height= "wrap_content"
    android:background= "@color/card_bg_green"
    android:orientation= "vertical">
    <LinearLayout
        android:layout_width= "match_parent"
```

```
    android:layout_height= "wrap_content"
    android:orientation= "horizontal"
    android:paddingLeft= "10dp"
    android:paddingRight= "10dp"
    android:paddingTop= "10dp">>

<TextView
    android:id= "@+id/sensor_name_text"
    style= "@style/FontSize"
    android:layout_width= "0dp"
    android:layout_height= "wrap_content"
    android:layout_weight= "1"
    android:gravity= "start/center_horizontal"
    android:text= "@string/light_title"
    android:textColor= "@color/white"
    android:textStyle= "bold"/>

<TextView
    android:id= "@+id/status_text"
    style= "@style/FontSize"
    android:layout_width= "wrap_content"
    android:layout_height= "wrap_content"
    android:gravity= "end"
    android:text= "@string/warning"
    android:textColor= "@color/white"
    android:textStyle= "bold"/>
</LinearLayout>
<LinearLayout
    android:layout_width= "match_parent"
    android:layout_height= "wrap_content"
    android:orientation= "horizontal"
    android:paddingBottom= "10dp"
    android:paddingLeft= "10dp"
    android:paddingRight= "10dp">

<TextView
    android:id= "@+id/set_value_text"
    style= "@style/FontSize"
    android:layout_width= "0dp"
    android:layout_height= "wrap_content"
    android:layout_weight= "1"
```

```
        android:gravity= "start/center_horizontal"
        android:text= "@string/set_value"
        android:textColor= "@color/white"/>

    <TextView
        android:id= "@+id/sensor_value_text"
        android:layout_width= "120dp"
        android:layout_height= "120dp"
        android:layout_gravity= "end"
        android:layout_marginRight= "10dp"
        android:background= "@drawable/sensor_value_bg"
        android:gravity= "center"
        android:text= "@string/value"
        android:textColor= "@color/white"
        android:textSize= "35sp"
        android:textStyle= "bold"/>
    </LinearLayout>
</LinearLayout>

</LinearLayout>
(7)res/values/array.xml 中的核心代码
<? xml version= "1.0" encoding= "utf-8"? >
<resources>

    <string-array name= "tag">
        <item name= "envir">环境指标</item>
        <item name= "history">历史查询</item>
        <item name= "control">手动控制</item>
        <item name= "setting">系统设置</item>
    </string-array>
    <string-array name= "sensor">
        <item name= "co2">CO2</item>
        <item name= "light">光照</item>
        <item name= "air_h">空气湿度</item>
        <item name= "air_t">空气温度</item>
        <item name= "soil_h">土壤湿度</item>
        <item name= "soil_t">土壤温度</item>
    </string-array>
</resources>
```

(8) res/values/colors.xml 中的核心代码

```
<? xml version= "1.0" encoding= "utf-8"? >
<resources>

<color name= "white"> # ffffffff</color>
<color name= "card_bg_green"> # bb22ee00</color>
<color name= "card_bg_red"> # bbf0000</color>
<color name= "cd"> # cdcacd</color>

</resources>
```

(9) res/values/strings.xml 中的核心代码

```
<? xml version= "1.0" encoding= "utf-8"? >
<resources>

<string name= "app_name">MainLayout</string>
<string name= "light_title">Light</string>
<string name= "set_value">设定值:</string>
<string name= "value">8888</string>
<string name= "warning">预警</string>
<string name= "normal">正常</string>

</resources>
```

(10) res/values/styles.xml 中的核心代码

```
<? xml version= "1.0" encoding= "utf-8"? >
<resources>

<style name= "AppBaseTheme" parent= "android:Theme.Light">
</style>
<style name= "AppTheme" parent= "AppBaseTheme">
</style>
<style name= "FontSize" parent= "AppBaseTheme">
    <item name= "android:textSize">25sp</item>
</style>

</resources>
```

(11) AndroidManifest.xml 中的核心代码

```
<? xml version= "1.0" encoding= "utf-8"? >
<manifest xmlns:android= "http://schemas.android.com/apk/res/android"
    package= "lenovo.agriculture.mainlayout"
    android:versionCode= "1"
    android:versionName= "1.0" >
```

```

<uses-sdk
    android:minSdkVersion= "15"
    android:targetSdkVersion= "21"/>

<application
    android:allowBackup= "true"
    android:icon= "@drawable/ic_launcher"
    android:label= "@string/app_name"
    android:theme= "@style/AppTheme">
    <activity
        android:name= ".activity.MainActivity"
        android:label= "@string/app_name"
        android:screenOrientation= "landscape">
        <intent-filter>
            <action android:name= "android.intent.action.MAIN"/>

            <category android:name= "android.intent.category.LAUNCHER"/>
        </intent-filter>
    </activity>
</application>

</manifest>

```

4.2.4 任务运作(Operate)

开发工程师之间两两交叉代码评审，并根据评审纪要重构代码。

所有团队成员召开评审会，并填写表 4-2-2。

表 4-2-2

评审建议表

序号	建议描述	是否接纳	原因
1		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
2		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
3		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
4		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
5		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
6		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
7		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
8		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
9		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
10		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	

所有团队成员召开复盘会，并填写表 4-2-3。

表 4-2-3

复盘会总结表

回顾目标		评估结果	
当初的目的是什么 (期望的结果)		Highlights (与原来目标比)	
要达成的目标 & 路标		Lowlights (与原来目标比)	
Goal	1	Result	2
总结规律	Insight		3
经验 & 规律 (不要轻易下结论)	Analysis		4
行动计划	分析原因		
新举措:	成功关键因素 (主观/客观)		
叫停:	失败根本原因 (主观/客观)		
继续:			

同学们,根据塔克曼阶梯理论,你们的团队目前处于哪个阶段呢?如果还没有进入成熟阶段,那就让我们一起努力吧,在接下来的任务中,更加注重相互配合和协作,及时发现并改正团队配合过程中出现的问题,早日让我们的团队进入成熟阶段!

4.2.5 任务扩展(Extend)

课后练习:将该示例程序中界面下方的四个 Tab 放置在界面上方。

子任务 3 实现客户端与服务端的网络连接

- 任务目标:

 - 实现客户端与服务端的网络连接

- 课时分配:

 - 8 课时

4.3.1 任务构思(Conceive)

绝大多数商业 App 都包含连接到后台服务端的网络操作,联想智能农业系统也不例外。联想智能农业系统中需要网络连接的操作包括:用户登录注册、实时读取沙盘上传感器的数据、控制沙盘上的受控设备等。

本任务对应的示例程序的运行效果如下:

(1)启动示例程序,切换到获取数据界面,如图 4-3-1 所示。



图 4-3-1 获取数据界面

(2)输入处于同一个局域网内的服务端 App 的 IP 地址,如图 4-3-2 所示。



图 4-3-2 输入 IP 地址

(3)关闭软键盘,点击按钮【获取数据】,如图 4-3-3 和图 4-3-4 所示。

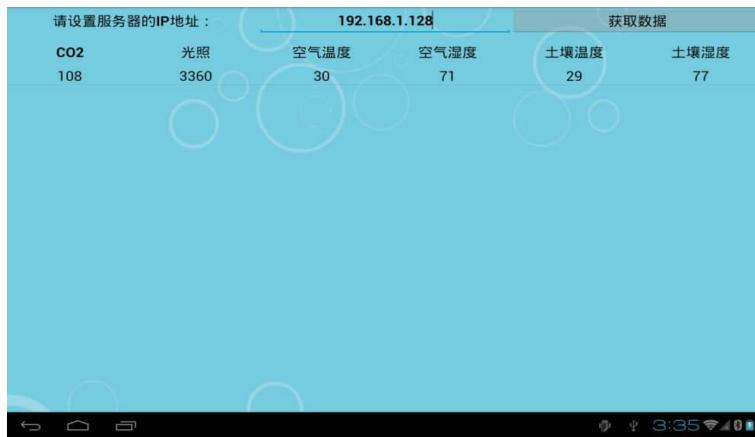


图 4-3-3 获取数据 1

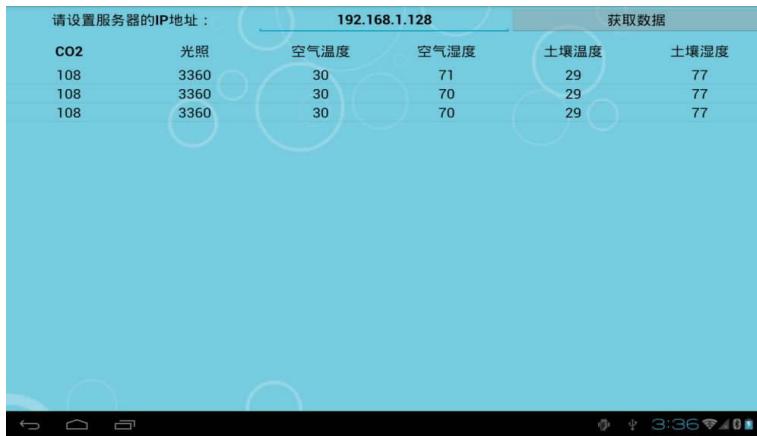


图 4-3-4 获取数据 2

4.3.2 任务设计(Design)

Android 提供了多种实现网络连接的方式,比如: Socket、HttpURLConnection、HttpClient、Web Service 等。本任务对应的示例程序使用的是 HttpURLConnection 这个 API。

4.3.3 任务实现(Implement)

所有团队成员召开计划会议,并将会议纪要记录在表 4-3-1 中。

表 4-3-1

会议纪要记录表

5W2H	结论
WHAT	
HOW	
WHY	
WHEN	
WHERE	
WHO	
HOW MUCH	

在计划会议结束之前,所有团队成员把子任务进一步细分为更小的任务,把这些任务写在便签上,并把便签贴在看板上的“ToDo”列。

所有团队成员每天早上召开 15 分钟的站立会议,每人汇报三个问题,并更新看板上的“ToDo”、“Doing”和“Done”三列。

本任务对应示例程序的核心代码如下:

(1) lenovo.agriculture.netconnection.util.NetUtil 类中的核心代码

```
/* *
 * 发送数据
```

```
* @param urlString 网络连接 Url
* @param params 发送的数据
* @return 服务器返回的数据
*/
public static String sendData(String urlString, String params) {
    String result = "";
    try {
        createConnection(urlString);
        setParams();
        writeData(params);
        result = readData(result);
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        if (mReader != null) {
            try {
                mReader.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
    return result;
}
/* *
 * 读取数据
 *
 * @param result 储存数据的字符串
 * @return 数据包
 */
private static String readData(String result) {
    try {
        mReader = new BufferedReader(new InputStreamReader(
            mConnection.getInputStream()));
        String line;
        while ((line = mReader.readLine()) != null) {
            if (result.equals("")) {
                result += line;
            } else {
                result += "\n" + line;
            }
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
    return result;
}
```

```
        }
    }

} catch (IOException e) {
    e.printStackTrace();
}

return result;
}

/*
 * 写出数据
 *
 * @param params 将要写出的数据
 */
private static void writeData(String params) {
    try {
        OutputStream os = mConnection.getOutputStream();
        OutputStreamWriter osw = new OutputStreamWriter(os, "utf-8");
        osw.write(params);
        osw.flush();
        osw.close();
    } catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

/*
 * 设置网络连接的参数
 */
private static void setParams() {
    mConnection.setConnectTimeout(20 * 1000);
    mConnection.setReadTimeout(20 * 1000);
    mConnection.setDoOutput(true);
    mConnection.setDoInput(true);
}

/*
 * 创建网络连接
 *
 * @param urlString 网络连接的 Url
 */
private static void createConnection(String urlString) {
    try {
```

```
mUrl = new URL(urlString);
mConnection = (HttpURLConnection) mUrl.openConnection();
} catch (MalformedURLException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
}

}

(2)res/layout/activity_main.xml 中的核心代码
<? xml version= "1.0" encoding= "utf-8"? >
<RelativeLayout xmlns:android= "http://schemas.android.com/apk/res/android"
    xmlns:tools= "http://schemas.android.com/tools"
    android:layout_width= "match_parent"
    android:layout_height= "match_parent"
    android:background= "@drawable/app_bg">

    <LinearLayout
        android:id= "@+id/linearLayout1"
        android:layout_width= "match_parent"
        android:layout_height= "wrap_content"
        android:layout_below= "@+id/linearLayout3"
        android:layout_marginTop= "16dp">

        <TextView
            android:id= "@+id/co2_tv"
            style= "@style/TitleStyle"
            android:text= "@string/co2"/>

        <TextView
            android:id= "@+id/light_tv"
            style= "@style/TitleStyle"
            android:text= "@string/light"/>

        <TextView
            android:id= "@+id/airt_t_tv"
            style= "@style/TitleStyle"
            android:text= "@string/air_t"/>

        <TextView
            android:id= "@+id/air_h_tv"
            style= "@style/TitleStyle"
```

```
    android:text= "@+string/air_h"/>

<TextView
    android:id= "@+id/soil_t_tv"
    style= "@style/TitleStyle"
    android:text= "@+string/soil_t"/>

<TextView
    android:id= "@+id/soil_h_tv"
    style= "@style/TitleStyle"
    android:text= "@+string/soil_h"/>
</LinearLayout>

<ListView
    android:id= "@+id/data_show_listview"
    android:layout_width= "match_parent"
    android:layout_height= "wrap_content"
    android:layout_alignParentBottom= "true"
    android:layout_below= "@+id/linearLayout1"
    android:layout_marginTop= "10dp">
</ListView>

<LinearLayout
    android:id= "@+id/linearLayout3"
    android:layout_width= "match_parent"
    android:layout_height= "wrap_content"
    android:orientation= "horizontal">

<TextView
    android:id= "@+id/textView2"
    style= "@style/TitleStyle"
    android:text= "@+string/ip_set"/>

<EditText
    android:id= "@+id/ip_ed"
    style= "@style/TitleStyle"
    android:ems= "10"
    android:hint= "@+string/ip_msg">

    <requestFocus/>
</EditText>
```

```
<Button
    android:id="@+id/get_data_btn"
    style="@style/TitleStyle"
    android:text="@string/get_data"/>
</LinearLayout>

</RelativeLayout>
(3)res/layout/data_item.xml 中的核心代码
<? xml version="1.0" encoding="utf-8"? >
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <TextView
        android:id="@+id/co2_tv"
        style="@style/ContentStyle"/>

    <TextView
        android:id="@+id/light_tv"
        style="@style/ContentStyle"/>

    <TextView
        android:id="@+id/air_t_tv"
        style="@style/ContentStyle"/>

    <TextView
        android:id="@+id/air_h_tv"
        style="@style/ContentStyle"/>

    <TextView
        android:id="@+id/soil_t_tv"
        style="@style/ContentStyle"/>

    <TextView
        android:id="@+id/soil_h_tv"
        style="@style/ContentStyle"/>

</LinearLayout>
```

(4) res/values/strings.xml 中的核心代码

```
<? xml version= "1.0" encoding= "utf-8"? >
<resources>

    <string name= "app_name">NetConnection</string>
    <string name= "co2">CO2</string>
    <string name= "light">光照</string>
    <string name= "air_h">空气湿度</string>
    <string name= "air_t">空气温度</string>
    <string name= "soil_h">土壤湿度</string>
    <string name= "soil_t">土壤温度</string>
    <string name= "ip_set">请设置服务器的 IP 地址:</string>
    <string name= "get_data">获取数据</string>
    <string name= "net_msg">当前未连接网络,请连接网络再试! </string>
    <string name= "ip_msg">请先填写服务器的 IP</string>
    <string name= "get_data_msg">数据获取中。。。</string>

</resources>
```

(5) res/values/styles.xml 中的核心代码

```
<? xml version= "1.0" encoding= "utf-8"? >
<resources>

    <style name= "AppBaseTheme" parent= "android:Theme.Light">
        </style>

    <style name= "AppTheme" parent= "AppBaseTheme">
        </style>

    <style name= "TitleStyle" parent= "AppBaseTheme">
        <item name= "android:layout_width">0dp</item>
        <item name= "android:layout_height">match_parent</item>
        <item name= "android:layout_weight">1</item>
        <item name= "android:textSize">25sp</item>
        <item name= "android:textStyle">bold</item>
        <item name= "android:gravity">center</item>
        <item name= "android:paddingLeft">5dp</item>
        <item name= "android:paddingRight">5dp</item>
    </style>

    <style name= "ContentStyle" parent= "AppBaseTheme">
        <item name= "android:layout_width">0dp</item>
```

```
<item name= "android:layout_height">match_parent</item>
<item name= "android:layout_weight">1</item>
<item name= "android:textSize">25sp</item>
<item name= "android:gravity">center</item>
<item name= "android:paddingLeft">5dp</item>
<item name= "android:paddingRight">5dp</item>
</style>

</resources>

(6)AndroidManifest.xml 中的核心代码
<? xml version= "1.0" encoding= "utf-8"? >
<manifest xmlns:android= "http://schemas.android.com/apk/res/android"
    package= "lenovo.agriculture.netconnection"
    android:versionCode= "1"
    android:versionName= "1.0" >

    <uses-sdk
        android:minSdkVersion= "15"
        android:targetSdkVersion= "21"/>

    <uses-permission android:name= "android.permission.INTERNET"/>
    <uses-permission android:name= "android.permission.ACCESS_NETWORK_STATE"/>

    <application
        android:allowBackup= "true"
        android:icon= "@drawable/ic_launcher"
        android:label= "@string/app_name"
        android:theme= "@style/AppTheme">
        <activity
            android:name= ".activity.MainActivity"
            android:label= "@string/app_name"
            android:screenOrientation= "landscape">
            <intent-filter>
                <action android:name= "android.intent.action.MAIN"/>

                <category android:name= "android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
    </application>

</manifest>
```

4.3.4 任务运作(Operate)

开发工程师之间两两交叉代码评审，并根据评审纪要重构代码。

所有团队成员召开评审会，并填写表 4-3-2。

表 4-3-2

评审建议表

序号	建议描述	是否接纳	原因
1		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
2		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
3		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
4		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
5		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
6		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
7		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
8		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
9		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
10		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	

所有团队成员召开复盘会，并填写表 4-3-3。

表 4-3-3

复盘会总结表

回顾目标		评估结果	
当初的目的是什么（期望的结果）		Highlights (与原来目标比)	
要达成的目标 & 里程碑		Lowlights (与原来目标比)	
Goal	1	Result	2
Insight	3	Analysis	4
分析原因			
总结规律			
经验 & 规律（不要轻易下结论）			
行动计划			
新举措：			
叫停：			
继续：			

同学们，根据塔克曼阶梯理论，你们的团队目前处于哪个阶段呢？如果还没有进入成熟阶段，那就让我们一起努力吧，在接下来的任务中，更加注重相互配合和协作，及时

发现并改正团队配合过程中出现的问题,早日让我们的团队进入成熟阶段!

4.3.5 任务扩展(Extend)

课后练习:将该示例程序中的 HttpURLConnection 改为 HttpClient,实现客户端到服务端的连接。

子任务 4 实现用户的注册和登录

- 任务目标:

- 实现用户的注册和登录

- 课时分配:

- 4 课时

4.4.1 任务构思(Conceive)

绝大多数商业 App 都包含用户的注册和登录功能,联想智能农业系统也不例外。此外,在用户注册和登录之前,需要先在客户端设置要连接的服务端 App 的 IP 地址。

本任务对应的示例程序的运行效果如下:

(1)启动示例程序,进入注册登录界面,如图 4-4-1 所示。



图 4-4-1 注册登录界面

(2)点击右上角的按钮【IP 设置】,进入 IP 设置界面,设置要连接的服务端 App 的 IP 地址,如图 4-4-2 所示。



图 4-4-2 设置 IP 地址

(3)点击按钮【用户注册】，进入用户注册界面，如图 4-4-3 所示。

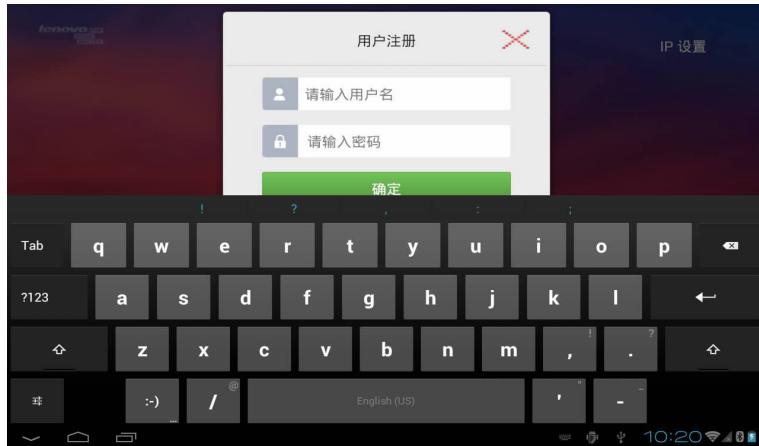


图 4-4-3 用户注册界面

(4)填写用户名和密码，如图 4-4-4 所示。



图 4-4-4 填写用户名和密码

(5)点击按钮【确定】，如果注册成功，如图 4-4-5 所示，否则，如图 4-4-6 所示。



图 4-4-5 注册成功

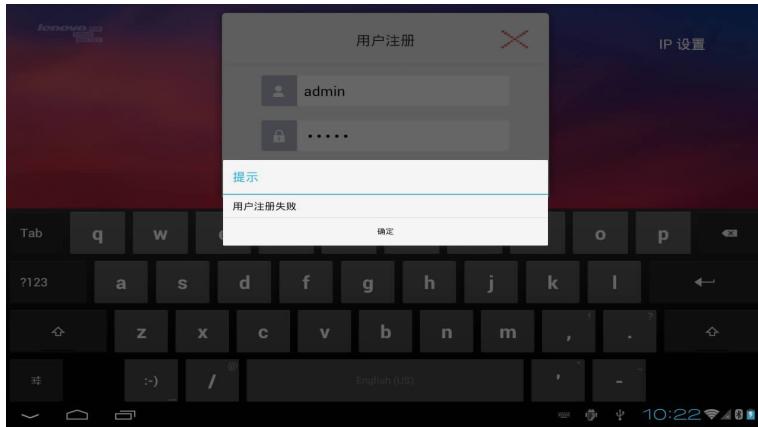


图 4-4-6 注册失败

(6) 注册成功后，在注册登录界面输入注册过的用户名及其对应的密码，如图 4-4-7 所示。点击按钮【登录】，如果成功登录，显示如图 4-4-8 所示的界面；否则，显示如图 4-4-9 所示的界面。



图 4-4-7 输入用户名及密码



图 4-4-8 登录成功



图 4-4-9 登录失败

4.4.2 任务设计(Design)

首先启动服务端 App，在服务端 App 的主界面显示其 IP 地址。然后启动客户端 App，并输入要连接到服务端 App 的 IP 地址，从而连接到指定的服务端。用户注册时，将设置的用户名和密码保存在服务端；用户登录时，根据输入的用户名和密码到服务端去查询，从而判断该用户是否是注册过的用户。

4.4.3 任务实现(Implement)

所有团队成员召开计划会议，并将会议纪要记录在表 4-4-1 中。

表 4-4-1

会议纪要记录表

5W2H	结论
WHAT	
HOW	
WHY	
WHEN	
WHERE	
WHO	
HOW MUCH	

在计划会议结束之前,所有团队成员把子任务进一步细分为更小的任务,把这些任务写在便签上,并把便签贴在看板上的“ToDo”列。

所有团队成员每天早上召开 15 分钟的站立会议,每人汇报三个问题,并更新看板上的“ToDo”、“Doing”和“Done”三列。

本任务对应示例程序的核心代码如下:

(1) lenovo.agriculture.userlogin.dialog.IpSetDialog 类中的核心代码

```
/*
 * 对用户输入的 IP 进行转化、校验和保存
 */
private void setIp() {
    try {
        // 获取 IP 地址
        int ip1, ip2, ip3, ip4;
        ip1 = Integer.parseInt(mIp1ET.getText().toString().trim());
        ip2 = Integer.parseInt(mIp2ET.getText().toString().trim());
        ip3 = Integer.parseInt(mIp3ET.getText().toString().trim());
        ip4 = Integer.parseInt(mIp4ET.getText().toString().trim());

        // 校验 IP 地址
        String ipStr = ip1 + "." + ip2 + "." + ip3 + "." + ip4;
        if (null == ipStr || "" .equals(ipStr)) {
            showAlertDialog(getString(R.string.prompt),
                getString(R.string.invalid_ip));
            return;
        } else {
            Pattern p = Pattern
```

第4章 项目实现（Implement）——编码和测试

```
.compile("(2[5][0-5]|2[0-4]\\d|1\\d{2}|\\d{1,2})\\.\\.(25[0-5]|2[0-4]\\d|1\\d{2}|\\d{1,2})\\.(25[0-5]|2[0-4]\\d|1\\d{2}|\\d{1,2})\\.(25[0-5]|2[0-4]\\d|1\\d{2}|\\d{1,2})");  
Matcher m = p.matcher(ipStr);  
if (!m.matches()) {  
    showAlertDialog(getString(R.string.prompt),  
        getString(R.string.invalid_ip));  
    return;  
}  
}  
mApp.setServerIpStr(ipStr);  
} catch (Exception e) {  
    showAlertDialog(getString(R.string.prompt),  
        getString(R.string.invalid_ip));  
}
```

(2) lenovo.agriculture.userlogin.dialog. UserRegisterDialog 类中的核心代码

```
/* *  
* 用户注册方法  
*/  
private void userRegisterEnter() {  
    String userName = mUserNameET.getText().toString().trim();  
    String password = mPasswordET.getText().toString().trim();  
    if (userName.equals("")) {  
        showAlertDialog(mContext.getString(R.string.prompt),  
            mContext.getString(R.string.username_not_empty));  
        return;  
    }  
    if (password.equals("")) {  
        showAlertDialog(mContext.getString(R.string.prompt),  
            mContext.getString(R.string.password_not_empty));  
        return;  
    }  
}
```

```
mRequest.setUserName(userName);  
mRequest.setPassword(password);  
startRequest(mRequest);  
}
```

(3) lenovo.agriculture.userlogin.activity. UserLoginActivity 类中的核心代码

```
/* *  
* 用户登录方法  
*/
```

```
* /  
private void userLoginEntry() {  
    String userName = mUserNameET.getText().toString().trim();  
    String password = mPasswordET.getText().toString().trim();  
    if (userName.equals("")) {  
        showAlertDialog(getString(R.string.prompt),  
            getString(R.string.username_not_empty));  
        return;  
    }  
    if (password.equals("")) {  
        showAlertDialog(getString(R.string.prompt),  
            getString(R.string.password_not_empty));  
        return;  
    }  
    mApp.setUsername(userName);  
    mRequest.setUserName(userName);  
    mRequest.setPassword(password);  
    setLoadingDialog();  
}
```

(4)res/layout/set_ip_dialog.xml 中的核心代码

```
<? xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:background="@drawable/dialog_bg"  
    android:orientation="vertical"  
    android:padding="30dp">  
  
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="0dp"  
    android:layout_weight="1">  
  
<TextView  
    style="@style/FontSize"  
    android:layout_width="0dp"  
    android:layout_height="match_parent"  
    android:layout_weight="6"  
    android:gravity="center"  
    android:text="@string/input_ip_addr"  
    android:textColor="#292929"/>
```

```
<Button  
    android:id="@+id/close"  
    android:layout_width="0dp"  
    android:layout_height="wrap_content"  
    android:layout_weight="1"  
    android:background="@drawable/dialog_close"/>  
</LinearLayout>  
  
<View  
    android:layout_width="wrap_content"  
    android:layout_height="1dp"  
    android:background="@drawable/dialog_line"/>  
  
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="0dp"  
    android:layout_weight="0.5"  
    android:gravity="center">  
  
<EditText  
    android:id="@+id/ipl_edit_text"  
    style="@style/FontSize"  
    android:layout_width="0dp"  
    android:layout_height="match_parent"  
    android:layout_marginLeft="5dp"  
    android:layout_weight="1"  
    android:gravity="center"  
    android:inputType="number"  
    android:maxLength="3"/>  
  
<EditText  
    android:id="@+id/ip2_edit_text"  
    style="@style/FontSize"  
    android:layout_width="0dp"  
    android:layout_height="match_parent"  
    android:layout_marginLeft="5dp"  
    android:layout_weight="1"  
    android:gravity="center"  
    android:inputType="number"
```

```
    android:maxLength= "3"/>

    <EditText
        android:id= "@+id/ip3_edit_text"
        style= "@style/FontSize"
        android:layout_width= "0dp"
        android:layout_height= "match_parent"
        android:layout_marginLeft= "5dp"
        android:layout_weight= "1"
        android:gravity= "center"
        android:inputType= "number"
        android:maxLength= "3"/>

    <EditText
        android:id= "@+id/ip4_edit_text"
        style= "@style/FontSize"
        android:layout_width= "0dp"
        android:layout_height= "match_parent"
        android:layout_marginLeft= "5dp"
        android:layout_weight= "1"
        android:gravity= "center"
        android:inputType= "number"
        android:maxLength= "3"/>
    </LinearLayout>

    <Button
        android:id= "@+id/ok_button"
        style= "@style/FontSize"
        android:layout_width= "500dp"
        android:layout_height= "0dp"
        android:layout_gravity= "center|bottom"
        android:layout_marginTop= "15dp"
        android:layout_weight= "0.5"
        android:background= "@drawable/btn_green"
        android:text= "@android:string/ok"
        android:textColor= "@android:color/white"/>

    </LinearLayout>
(5)res/layout/user_register_dialog.xml 中的核心代码
<? xml version= "1.0" encoding= "utf-8"? >
<LinearLayout xmlns:android= "http://schemas.android.com/apk/res/android"
```

```
    android:layout_width= "600dp"
    android:layout_height= "500dp"
    android:background= "@drawable/dialog_bg"
    android:orientation= "vertical"
    android:padding= "20dp">>

    <RelativeLayout
        android:layout_width= "match_parent"
        android:layout_height= "90dp">

        <TextView
            style= "@style/FontSize"
            android:layout_width= "wrap_content"
            android:layout_height= "wrap_content"
            android:layout_centerInParent= "true"
            android:text= "@string/register"
            android:textColor= "#292929"/>

        <Button
            android:id= "@+id/close"
            android:layout_width= "wrap_content"
            android:layout_height= "wrap_content"
            android:layout_alignParentRight= "true"
            android:layout_centerVertical= "true"
            android:layout_marginRight= "20dp"
            android:background= "@drawable/dialog_close"/>
    </RelativeLayout>

    <View
        android:layout_width= "wrap_content"
        android:layout_height= "1dp"
        android:background= "@drawable/dialog_line"/>

    <EditText
        android:id= "@+id/account_edit_text"
        style= "@style/FontSize"
        android:layout_width= "match_parent"
        android:layout_height= "60dp"
        android:layout_marginLeft= "60dp"
        android:layout_marginRight= "60dp"
        android:layout_marginTop= "20dp"
```

```
    android:background= "@drawable/username_edit_bg"
    android:hint= "@string/username_hint"
    android:singleLine= "true"/>

<EditText
    android:id= "@+id/password_edit_text"
    style= "@style/FontSize"
    android:layout_width= "match_parent"
    android:layout_height= "60dp"
    android:layout_marginLeft= "60dp"
    android:layout_marginRight= "60dp"
    android:layout_marginTop= "25dp"
    android:background= "@drawable/password_edit_bg"
    android:hint= "@string/password_hint"
    android:inputType= "textPassword"
    android:singleLine= "true"/>

<Button
    android:id= "@+id/ok_button"
    style= "@style/FontSize"
    android:layout_width= "match_parent"
    android:layout_height= "60dp"
    android:layout_marginLeft= "60dp"
    android:layout_marginRight= "60dp"
    android:layout_marginTop= "25dp"
    android:background= "@drawable/btn_green"
    android:text= "@android:string/ok"
    android:textColor= "@color/white"/>

</LinearLayout>
(6)res/layout/user_login.xml 中的核心代码
<? xml version= "1.0" encoding= "utf-8"? >
<LinearLayout xmlns:android= "http://schemas.android.com/apk/res/android"
    android:layout_width= "match_parent"
    android:layout_height= "match_parent"
    android:background= "@drawable/bg"
    android:gravity= "center_horizontal"
    android:orientation= "vertical">

<TextView
    android:id= "@+id/set_ip_text_view"
```

```
style="@style/FontSize"
    android:layout_width="150dp"
    android:layout_height="50dp"
    android:layout_marginLeft="500dp"
    android:layout_marginTop="50dp"
    android:gravity="center"
    android:text="@string/setting_ip"
    android:textColor="@color/white"/>

<LinearLayout
    android:layout_width="500dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="110dp"
    android:background="@drawable/dialog_bg"
    android:gravity="top/center_horizontal"
    android:orientation="vertical"
    android:padding="50dp">

<EditText
    android:id="@+id/account_edit_text"
    style="@style/FontSize"
    android:layout_width="match_parent"
    android:layout_height="60dp"
    android:background="@drawable/username_edit_bg"
    android:hint="@string/username_hint"
    android:singleLine="true"/>

<EditText
    android:id="@+id/password_edit_text"
    style="@style/FontSize"
    android:layout_width="match_parent"
    android:layout_height="60dp"
    android:layout_marginTop="25dp"
    android:background="@drawable/password_edit_bg"
    android:hint="@string/password_hint"
    android:inputType="textPassword"
    android:singleLine="true"/>

<Button
    android:id="@+id/login_button"
    style="@style/FontSize"
```

```
    android:layout_width= "match_parent"
    android:layout_height= "60dp"
    android:layout_marginBottom= "15dp"
    android:layout_marginTop= "25dp"
    android:background= "@drawable/btn_blue"
    android:text= "@string/login"
    android:textColor= "@color/white"/>

<TextView
    android:id= "@+id/register_text_view"
    style= "@style/FontSize"
    android:layout_width= "398dp"
    android:layout_height= "40dp"
    android:layout_gravity= "end/center_vertical"
    android:layout_marginLeft= "20dp"
    android:layout_weight= "0.08"
    android:background= "@drawable/blue_textview_bg"
    android:gravity= "center_vertical/center_horizontal"
    android:paddingLeft= "5dp"
    android:text= "@string/register"
    android:textColor= "@color/blue"/>
</LinearLayout>

</LinearLayout>
(7)res/values/colors.xml 的核心代码
<? xml version= "1.0" encoding= "utf-8"? >
<resources>

    <color name= "white">#ffffffff</color>
    <color name= "blue">#ff0022ff</color>

</resources>
(8)res/values/styles.xml 中的核心代码
<? xml version= "1.0" encoding= "utf-8"? >
<resources>

    <style name= "AppBaseTheme" parent= "android:Theme.Light">
    </style>
    <style name= "AppTheme" parent= "AppBaseTheme">
    </style>
    <style name= "FontSize" parent= "AppBaseTheme">
```

```
<item name= "android:textSize">25sp</item>
</style>
```

```
</resources>
```

(9) res/values/strings.xml 中的核心代码

```
<? xml version= "1.0" encoding= "utf-8"? >
<resources>
```

```
<string name= "app_name">UserLogin</string>
<string name= "login">登录</string>
<string name= "register">用户注册</string>
<string name= "setting_ip">IP 设置</string>
<string name= "failed">本次操作失败</string>
<string name= "login_failed">用户登录失败</string>
<string name= "register_success">用户注册成功</string>
<string name= "register_failed">用户注册失败</string>
<string name= "username_hint">请输入用户名</string>
<string name= "username_not_empty">用户名不能为空</string>
<string name= "password_hint">请输入密码</string>
<string name= "password_not_empty">密码不能为空</string>
<string name= "input_ip_addr">请输入 IP 地址</string>
<string name= "invalid_ip">无效的 IP 地址</string>
<string name= "prompt">提示</string>
<string name= "login_wait">登录中。。。</string>
<string name= "no_net_work">请连接网络</string>
```

```
</resources>
```

(10) AndroidManifest.xml 中的核心代码

```
<? xml version= "1.0" encoding= "utf-8"? >
```

```
<manifest xmlns:android= "http://schemas.android.com/apk/res/android"
```

```
    package= "lenovo.agriculture.userlogin"
    android:versionCode= "1"
    android:versionName= "1.0" >
```

```
<uses-sdk
    android:minSdkVersion= "15"
    android:targetSdkVersion= "21"/>
```

```
<uses-permission android:name= "android.permission.ACCESS_NETWORK_STATE" />
```

```
<uses-permission android:name= "android.permission.INTERNET" />
```

```
<uses-permission android:name= "android.permission.ACCESS_WIFI_STATE" />
```

```
<application
    android:name=".ClientApp"
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme">
    <activity
        android:name=".activity.UserLoginActivity"
        android:label="@string/app_name"
        android:screenOrientation="landscape"
        android:windowSoftInputMode="stateHidden/adjustUnspecified">
        <intent-filter>
            <action android:name="android.intent.action.MAIN"/>

            <category android:name="android.intent.category.LAUNCHER"/>
        </intent-filter>
    </activity>
    <activity android:name=".activity BaseActivity"/>
    <activity
        android:name=".activity.MainActivity"
        android:screenOrientation="landscape"/>
</application>

</manifest>
```

4.4.4 任务运作(Operate)

开发工程师之间两两交叉代码评审，并根据评审纪要重构代码。

所有团队成员召开评审会，并填写表 4-4-2。

表 4-4-2 评审建议表

序号	建议描述	是否接纳	原因
1		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
2		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
3		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
4		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
5		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
6		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	

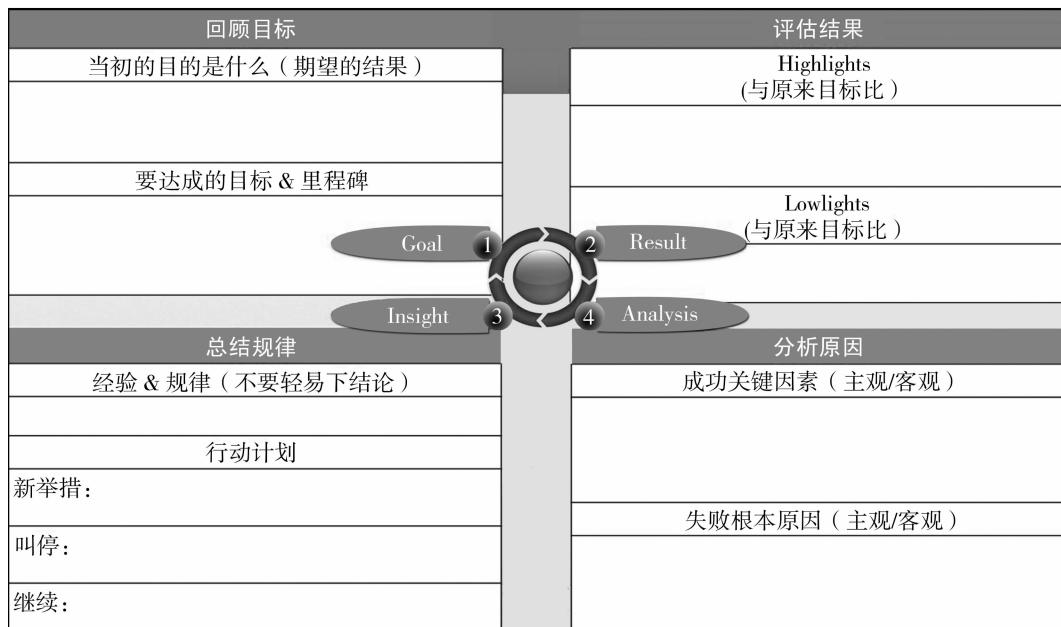
(续表)

序号	建议描述	是否接纳	原因
7		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
8		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
9		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
10		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	

所有团队成员召开复盘会，并填写表 4-4-3。

表 4-4-3

复盘会总结表



同学们，根据塔克曼阶梯理论，你们的团队目前处于哪个阶段呢？如果还没有进入成熟阶段，那就让我们一起努力吧，在接下来的任务中，更加注重相互配合和协作，及时发现并改正团队配合过程中出现的问题，早日让我们的团队进入成熟阶段！

4.4.5 任务扩展(Extend)

课后练习：在该示例程序的基础上，添加记住密码以便自动登录的功能。

子任务 5 实现客户端实时读取沙盘上传感器的数据

- 任务目标：

- 实现客户端实时读取沙盘上传感器的数据

- 课时分配：

4 课时

4.5.1 任务构思(Conceive)

在联想智能农业系统中，客户端 App 会每隔一秒实时读取沙盘上各个传感器的数据，并将数据显示在环境指标界面，以便用户实时查看沙盘上的环境参数。

本任务对应的示例程序的运行效果如下：

(1)启动示例程序，设置客户端 App 要连接的服务端的 IP 地址，如图 4-5-1 所示。



图 4-5-1 设置 IP 地址

(2)点击按钮【确定】进入环境指标界面，该界面的状态会随着沙盘中各个传感器的数据变化而不断变化，如图 4-5-2、图 4-5-3 和图 4-5-4 所示。



图 4-5-2 界面变化 1



图 4-5-3 界面变化 2



图 4-5-4 界面变化 3

4.5.2 任务设计(Design)

该任务的设计思路如下：

- (1) 客户端 App 通过 JSON 协议将获取数据的请求发送给服务端；
- (2) 服务端从沙盘上实时读取各个传感器的数据；
- (3) 服务端将读取的数据通过 JSON 协议返回给客户端 App。

4.5.3 任务实现(Implement)

所有团队成员召开计划会议，并将会议纪要记录在表 4-5-1 中。

表 4-5-1

会议纪要记录表

5W2H	结论
WHAT	
HOW	
WHY	
WHEN	
WHERE	
WHO	
HOW MUCH	

在计划会议结束之前,所有团队成员把子任务进一步细分为更小的任务,把这些任务写在便签上,并把便签贴在看板上的“ToDo”列。

所有团队成员每天早上召开 15 分钟的站立会议,每人汇报三个问题,并更新看板上的“ToDo”、“Doing”和“Done”三列。

本任务对应示例程序的核心代码如下:

(1)lenovo.agriculture.getsensordata.activity.MainActivity 类中的核心代码

```
/*
 * 设置、启动获取传感器当前值的线程
 */
private void setGetSenosorThread() {
    mSensorThread = new RequestThread(this, mApp.getHandler());
    mSensorThread.setLoop(true, 1500);
    mSensorRequest = new GetSensorRequest("");
    mSensorRequest.setSensorValue(mApp.getCurSensorValue());
    mSensorRequest
        .setOnResponseEventListener(new OnResponseEventListener() {


```

```
        @Override
        public void onResponse(BaseRequest request,
            RequestResult result) {
            mUpDate.upDate();
        }
    });
}
```

```
mSensorThread.setRequest(mSensorRequest);
mSensorThread.start();
}
```

(2)lenovo.agriculture.getsensordata.fragment.EnvirFragment 类中的核心代码

```
/*
 * 更新数据
*/
```

```
* /  
public void upDateView() {  
    mList.clear();  
    SensorBean mBean;  
    SensorValue mValue = mApp.getCurSensorValue();  
    SensorConfig mConfig = mApp.getSensorConfig();  
    // 设置 CO2 传感器的数据  
    mBean = new SensorBean(mStrings[0]);  
    mBean.setMaxValue(mConfig.maxCo2);  
    mBean.setMinValue(mConfig.minCo2);  
    mBean.setValue(mValue.getCo2());  
    mList.add(mBean);  
    // 设置光照传感器的数据  
    mBean = new SensorBean(mStrings[1]);  
    mBean.setMaxValue(mConfig.maxLight);  
    mBean.setMinValue(mConfig.minLight);  
    mBean.setValue(mValue.getLight());  
    mList.add(mBean);  
    // 设置空气温度传感器的数据  
    mBean = new SensorBean(mStrings[2]);  
    mBean.setMaxValue(mConfig.maxAirTemperature);  
    mBean.setMinValue(mConfig.minAirTemperature);  
    mBean.setValue(mValue.getAirTemper());  
    mList.add(mBean);  
    // 设置空气湿度传感器的数据  
    mBean = new SensorBean(mStrings[3]);  
    mBean.setMaxValue(mConfig.maxAirHumidity);  
    mBean.setMinValue(mConfig.minAirHumidity);  
    mBean.setValue(mValue.getAirHumid());  
    mList.add(mBean);  
    // 设置土壤温度传感器的数据  
    mBean = new SensorBean(mStrings[4]);  
    mBean.setMaxValue(mConfig.maxSoilTemperature);  
    mBean.setMinValue(mConfig.minSoilTemperature);  
    mBean.setValue(mValue.getSoilTemper());  
    mList.add(mBean);  
    // 设置土壤湿度传感器的数据  
    mBean = new SensorBean(mStrings[5]);  
    mBean.setMaxValue(mConfig.maxSoilHumidity);  
    mBean.setMinValue(mConfig.minSoilHumidity);  
    mBean.setValue(mValue.getSoilHumid());  
}
```

```
mList.add(mBean);
mAdapter.notifyDataSetChanged();
}

(3)res/layout/sensor_grid_item.xml 中的核心代码
<? xml version="1.0" encoding="utf-8"? >
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/card_bg_green"
    android:orientation="vertical">

    <LinearLayout
        android:id="@+id/sensor_item_layout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@color/card_bg_green"
        android:orientation="vertical">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal"
            android:paddingLeft="10dp"
            android:paddingRight="10dp"
            android:paddingTop="10dp">

            <TextView
                android:id="@+id/sensor_name_text"
                style="@style/FontSize"
                android:layout_width="0dp"
                android:layout_height="wrap_content"
                android:layout_weight="1"
                android:gravity="start/center_horizontal"
                android:text="@string/light_title"
                android:textColor="@color/white"
                android:textStyle="bold"/>

            <TextView
                android:id="@+id/status_text"
                style="@style/FontSize"
                android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
    android:gravity="end"
    android:text="@string/warning"
    android:textColor="@color/white"
    android:textStyle="bold"/>
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    android:paddingBottom="10dp"
    android:paddingLeft="10dp"
    android:paddingRight="10dp">

    <TextView
        android:id="@+id/set_value_text"
        style="@style/FontSize"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:gravity="start/center_horizontal"
        android:text="@string/set_value"
        android:textColor="@color/white"/>

    <TextView
        android:id="@+id/sensor_value_text"
        android:layout_width="120dp"
        android:layout_height="120dp"
        android:layout_gravity="end"
        android:background="@drawable/sensor_value_bg"
        android:gravity="center"
        android:text="@string/value"
        android:textColor="@color/white"
        android:textSize="35sp"
        android:textStyle="bold"/>
</LinearLayout>
</LinearLayout>

</LinearLayout>
```

(4)res/values/strings.xml 中的核心代码

```
<? xml version= "1.0" encoding= "utf-8"? >
<resources>

    <string name= "app_name">GetSensorData</string>
    <string name= "light_title">Light</string>
    <string name= "set_value">设定值:</string>
    <string name= "value">8888</string>
    <string name= "warning">预警</string>
    <string name= "normal">正常</string>
    <string name= "input_ip_addr">请输入 IP 地址</string>
    <string name= "invalid_ip">无效的 IP 地址</string>
    <string name= "prompt">提示</string>
    <string name= "no_net_work">请连接网络</string>

</resources>
```

(5)res/values/styles.xml 中的核心代码

```
<? xml version= "1.0" encoding= "utf-8"? >
<resources>

    <style name= "AppBaseTheme" parent= "android:Theme.Light">
        </style>
    <style name= "AppTheme" parent= "AppBaseTheme">
        </style>
    <style name= "FontSize" parent= "AppBaseTheme">
        <item name= "android:textSize">25sp</item>
    </style>

</resources>
```

(6)AndroidManifest.xml 的核心代码

```
<? xml version= "1.0" encoding= "utf-8"? >
<manifest xmlns:android= "http://schemas.android.com/apk/res/android"
    package= "lenovo.agriculture.getsensordata"
    android:versionCode= "1"
    android:versionName= "1.0" >

    <uses-sdk
        android:minSdkVersion= "15"
        android:targetSdkVersion= "21"/>

    <uses-permission android:name= "android.permission.ACCESS_WIFI_STATE"/>
```

```

<uses-permission android:name= "android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name= "android.permission.INTERNET"/>

<application
    android:name= ".ClientApp"
    android:allowBackup= "true"
    android:icon= "@drawable/ic_launcher"
    android:label= "@string/app_name"
    android:theme= "@style/AppTheme">
    <activity
        android:name= ".activity.MainActivity"
        android:label= "@string/app_name"
        android:screenOrientation= "landscape">
        <intent-filter>
            <action android:name= "android.intent.action.MAIN"/>

            <category android:name= "android.intent.category.LAUNCHER"/>
        </intent-filter>
    </activity>
</application>

</manifest>

```

4.5.4 任务运作(Operate)

开发工程师之间两两交叉代码评审，并根据评审纪要重构代码。

所有团队成员召开评审会，并填写表 4-5-2。

表 4-5-2

评审建议表

序号	建议描述	是否接纳	原因
1		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
2		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
3		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
4		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
5		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
6		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
7		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
8		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
9		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
10		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	

所有团队成员召开复盘会，并填写表 4-5-3。

表 4-5-3

复盘会总结表

回顾目标		评估结果	
当初的目的是什么（期望的结果）		Highlights (与原来目标比)	
要达成的目标 & 路标		Lowlights (与原来目标比)	
Goal		Result	
Insight		Analysis	
总结规律		分析原因	
经验 & 规律（不要轻易下结论）		成功关键因素（主观/客观）	
行动计划		失败根本原因（主观/客观）	
新举措：			
叫停：			
继续：			

同学们，根据塔克曼阶梯理论，你们的团队目前处于哪个阶段呢？如果还没有进入成熟阶段，那就让我们一起努力吧，在接下来的任务中，更加注重相互配合和协作，及时发现并改正团队配合过程中出现的问题，早日让我们的团队进入成熟阶段！

4.5.5 任务扩展(Extend)

课后练习：在该任务对应示例程序的基础上，使用 XML 文件替换 JSON 来封装数据。

子任务 6 在任务栏生成警告通知

- 任务目标：

 - 在任务栏生成警告通知

- 课时分配：

 - 4 课时

4.6.1 任务构思(Conceive)

在联想智能农业系统中，当沙盘上某个传感器的数据不在用户设置的阀值范围内时，就会在客户端 App 的状态栏生成警告通知，例如：空气温度过高、二氧化碳浓度过低等。

本任务对应示例程序的运行效果如下：

(1) 启动示例程序，进入主界面，如图 4-6-1 所示。

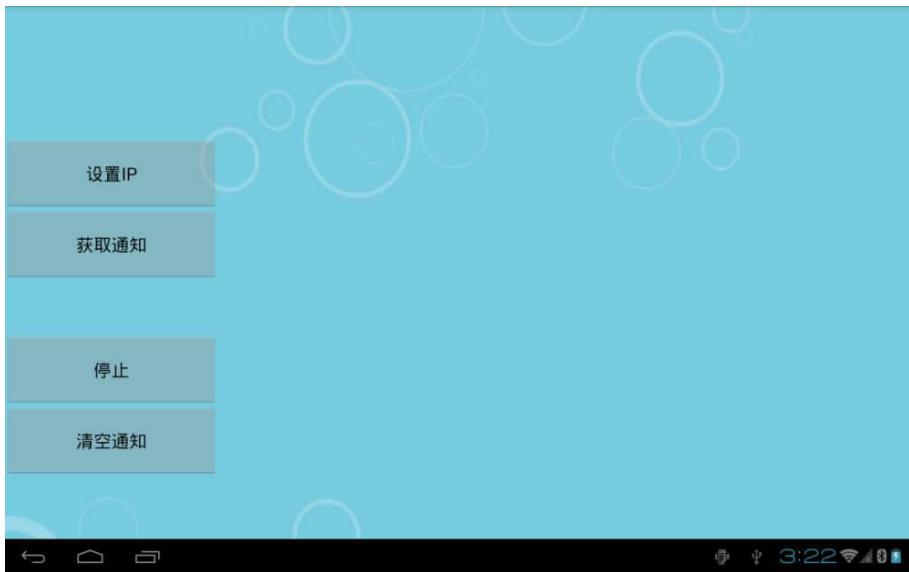


图 4-6-1 程序主界面

(2) 点击按钮【设置 IP】，设置客户端 App 要连接的服务端的 IP 地址，如图 4-6-2 所示。



图 4-6-2 设置 IP 地址

(3) 点击按钮【确定】后返回主界面。点击按钮【获取通知】，开启后台线程获取服务端发送的通知并显示在通知栏，如图 4-6-3 所示。



图 4-6-3 获取通知并显示在通知栏

(4)点击【停止】按钮,停止获取通知。

(5)点击按钮【清空通知】,将通知栏中的通知全部清空,如图 4-6-4 所示。

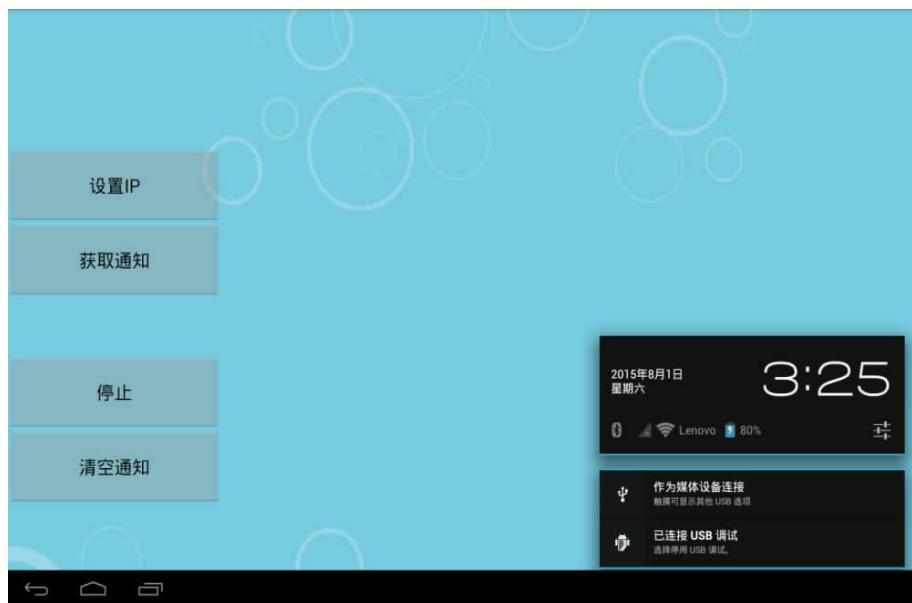


图 4-6-4 清空通知栏中的全部通知

4.6.2 任务设计(Design)

当服务端向客户端发出生成通知的命令时,使用系统控件 Notification 在客户端 App 的状态栏生成各个传感器对应的通知。

4.6.3 任务实现 (Implement)

所有团队成员召开计划会议，并将会议纪要记录在表 4-6-1 中。

表 4-6-1

会议纪要记录表

5W2H	结论
WHAT	
HOW	
WHY	
WHEN	
WHERE	
WHO	
HOW MUCH	

在计划会议结束之前，所有团队成员把子任务进一步细分为更小的任务，把这些任务写在便签上，并把便签贴在看板上的“ToDo”列。

所有团队成员每天早上召开 15 分钟的站立会议，每人汇报三个问题，并更新看板上的“ToDo”、“Doing”和“Done”三列。

本任务对应示例程序的核心代码如下：

(1) lenovo.agriculture.getnotification.service.NotifiService 类中的核心代码

```
/*
 * 处理通知
 * @param notifiStr 通知内容
 */
private void handlerNotifiStr(String notifiStr) {
    if (notifiStr == null || notifiStr.equals("")) {
        return;
    }
    long id = -1;
    String notifiMsg = "";
    try {
        JSONObject jsonRespObj = new JSONObject(notifiStr);
        if (jsonRespObj != null) {
            if (jsonRespObj.has("notifiId")) {
                id = jsonRespObj.getLong("notifiId");
            }
            if (jsonRespObj.has("notifiMsg")) {
                notifiMsg = jsonRespObj.getString("notifiMsg");
            }
        }
    } catch (JSONException e) {
        e.printStackTrace();
    }
}
```

```
        }

    }

} catch (JSONException e) {
    e.printStackTrace();
}

if (id > -1 && id != mNotifiId) {
    mNotifiId = id;
    setNotification(notifiMsg);
}

}

/*
 * 设置通知
 * @param notifiMsg 通知消息
 */

private void setNotification(String notifiMsg) {
    if (notifiMsg == null) {
        return;
    }

    NotificationManager mNotificationManager=
        (NotificationManager) getSystemService(Context.NOTIFICATION_SERVICE );
    for (int i = 0; i < mList.size(); i++) {
        if ((Boolean) ((Map<?, ? >) mList.get(i)).get("isFirst")) {
            mNotificationManager.cancel(i);
            mMap = (Map<String, Object>) mList.get(i);
            mMap.put("num", 0);
            mMap.put("isFirst", true );
        }
    }

    String notifis[] = notifiMsg.split(",");
    for (int j = 0; j < notifis.length; j++) {
        for (int i = 0; i < mList.size(); i++) {
            int num = Integer.valueOf(((Map<?, ? >) mList.get(i)).get("num")
                .toString());
            int mId = Integer.valueOf(((Map<?, ? >) mList.get(i)).get("id")
                .toString());
            if (notifis[j].equals(((Map<?, ? >) mList.get(i)).get("name"))) {
                showNotificationView(getString(ids[i - 1]), mId, ++num);
                mMap = (Map<String, Object>) mList.get(i);
                mMap.put("num", String.valueOf(num));
            }
        }
    }
}
```

```

        mMap.put("isFirst", false );
        break ;
    }
}
}
}
}
/*
* 创建通知
* @param message 通知内容
* @param id 通知 Id
*/
private void showNotificationView(String message, int id, int num) {
    NotificationCompat.Builder mBuilder = new NotificationCompat.Builder(
        this ).setSmallIcon( R.drawable.ic_launcher )
        .setContentTitle( getString( R.string.intelligent_agriculture ) )
        .setAutoCancel( true ).setContentText( message )
        .setWhen( System.currentTimeMillis() ).setNumber( num )
        .setDefaults( Notification.DEFAULT_ALL );
    Intent resultIntent = new Intent( this , MainActivity.class );
    TaskStackBuilder stackBuilder = TaskStackBuilder.create( this );
    stackBuilder.addParentStack( MainActivity.class );
    stackBuilder.addNextIntent( resultIntent );
    PendingIntent resultPendingIntent = stackBuilder.getPendingIntent( 0 ,
        PendingIntent.FLAG_UPDATE_CURRENT );
    mBuilder.setContentIntent( resultPendingIntent );
    NotificationManager mNotificationManager=
        (NotificationManager) getSystemService( Context.NOTIFICATION_SERVICE );
    mNotificationManager.notify( id, mBuilder.build() );
}

(2)lenovo.agriculture.getnotification.activity.MainActivity类中的核心代码
/*
* 清除所有通知
* @param v View
*/
public void clear( View v ) {
    NotificationManager manager =
        (NotificationManager) getSystemService( Context.NOTIFICATION_SERVICE );
    manager.cancelAll();
}

```

(3)res/layout/activity_main.xml 中的核心代码

```
<? xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/app_bg"
    android:orientation="horizontal">

    <Button
        android:id="@+id/button3"
        style="@style/FontSize"
        android:layout_width="300dp"
        android:layout_height="100dp"
        android:layout_alignParentBottom="true"
        android:layout_alignParentLeft="true"
        android:layout_marginBottom="89dp"
        android:onClick="clear"
        android:text="@string/clear"/>

    <Button
        android:id="@+id/button4"
        style="@style/FontSize"
        android:layout_width="300dp"
        android:layout_height="100dp"
        android:layout_above="@+id/button3"
        android:layout_alignParentLeft="true"
        android:onClick="stop"
        android:text="@string/stop"/>

    <Button
        android:id="@+id/button1"
        style="@style/FontSize"
        android:layout_width="300dp"
        android:layout_height="100dp"
        android:layout_above="@+id/button4"
        android:layout_alignParentLeft="true"
        android:layout_marginBottom="77dp"
        android:onClick="getNotifi"
        android:text="@string/get_notifi"/>
```

```
<Button
    android:id="@+id/button2"
    style="@style/FontSize"
    android:layout_width="300dp"
    android:layout_height="100dp"
    android:layout_above="@+id/button1"
    android:layout_alignParentLeft="true"
    android:onClick="setIp"
    android:text="@string/set_ip"/>
```

</RelativeLayout>

(4) res/values/strings.xml 中的核心代码

```
<? xml version="1.0" encoding="utf-8"? >
<resources>

    <string name="app_name">GetNotification</string>
    <string name="intelligent_ariculture">智能农业系统</string>
    <string name="pm25_high">PM2.5 浓度过高</string>
    <string name="co2_high">CO2 浓度过高</string>
    <string name="co2_low">CO2 浓度过低</string>
    <string name="light_high">光照强度过高</string>
    <string name="light_low">光照强度过低</string>
    <string name="air_tmper_high">空气温度过高</string>
    <string name="air_tmper_low">空气温度过低</string>
    <string name="air_humid_high">空气湿度过高</string>
    <string name="air_humid_low">空气湿度过低</string>
    <string name="soil_tmper_high">土壤湿度过高</string>
    <string name="soil_tmper_low">土壤湿度过低</string>
    <string name="soil_humid_high">土壤湿度过高</string>
    <string name="soil_humid_low">土壤湿度过低</string>
    <string name="input_ip_addr">请输入 IP 地址</string>
    <string name="invalid_ip">无效的 IP 地址</string>
    <string name="prompt">提示</string>
    <string name="set_ip">设置 IP</string>
    <string name="get_notifi">获取通知</string>
    <string name="stop">停止</string>
    <string name="clear">清空通知</string>
```

</resources>

(5) AndroidManifest.xml 中的核心代码

```
<? xml version="1.0" encoding="utf-8"? >
```

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="lenovo.agriculture.getnotification"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="15"
        android:targetSdkVersion="21"/>

    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.INTERNET"/>

    <application
        android:name=".ClientApp"
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme">
        <activity
            android:name="lenovo.agriculture.getnotification.activity.MainActivity"
            android:label="@string/app_name"
            android:screenOrientation="landscape">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>

                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>

        <service android:name=".service.NotifiService">
        </service>
    </application>

</manifest>
```

4.6.4 任务运作(Operate)

开发工程师之间两两交叉代码评审，并根据评审纪要重构代码。
所有团队成员召开评审会，并填写表 4-6-2。

表 4-6-2

评审建议表

序号	建议描述	是否接纳	原因
1		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
2		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
3		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
4		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
5		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
6		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
7		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
8		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
9		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
10		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	

所有团队成员召开复盘会，并填写表 4-6-3。

表 4-6-3

复盘会总结表

回顾目标		评估结果	
当初的目的是什么（期望的结果）		Highlights (与原来目标比)	
要达成的目标 & 里程碑		Lowlights (与原来目标比)	
Goal	1	2	Result
Insight	3	4	Analysis
分析原因			
总结规律		成功关键因素（主观/客观）	
经验 & 规律（不要轻易下结论）			
行动计划			
新举措：		失败根本原因（主观/客观）	
叫停：			
继续：			

同学们，根据塔克曼阶梯理论，你们的团队目前处于哪个阶段呢？如果还没有进入成熟阶段，那就让我们一起努力吧，在接下来的任务中，更加注重相互配合和协作，及时发现并改正团队配合过程中出现的问题，早日让我们的团队进入成熟阶段！

4.6.5 任务扩展(Extend)

课后练习：在该任务对应示例程序的基础上，当显示某个传感器的通知时，为该通知添加一个图标。

子任务 7 手动打开或关闭沙盘上的受控设备

- 任务目标：

- 手动打开或关闭沙盘上的受控设备

- 课时分配：

- 4 课时

4.7.1 任务构思(Conceive)

在联想智能农业系统中，客户端 App 可以手动打开或关闭沙盘上的受控设备，包括：水泵、蜂鸣器、风扇、灯管等。

本任务对应示例程序的运行效果如下：

(1)启动示例程序，设置客户端 App 要连接的服务端的 IP 地址，如图 4-7-1 所示。



图 4-7-1 设置 IP 地址

(2)点击按钮【确定】，进入风扇的控制界面，如图 4-7-2 所示。



图 4-7-2 风扇控制界面

(3) 点击按钮【打开】即可打开风扇, 如图 4-7-3 所示。



图 4-7-3 打开风扇

(4) 点击按钮【关闭】即可关闭风扇, 如图 4-7-2 所示。

(5) 手指向右滑屏切换到光照控制界面, 如图 4-7-4 所示。



图 4-7-4 光照控制界面

(6) 点击按钮【打开】即可打开 LED 灯, 如图 4-7-5 所示。



图 4-7-5 打开 LED 灯

(7) 点击按钮【关闭】即可关闭 LED 灯, 如图 4-7-4 所示。

(8) 手指向右滑屏切换到水泵控制界面,如图 4-7-6 所示。



图 4-7-6 水泵控制界面

(9) 点击按钮【打开】即可打开水泵,如图 4-7-7 所示。



图 4-7-7 打开水泵

(10) 点击按钮【关闭】即可关闭水泵,如图 4-7-6 所示。

(11) 手指向右滑屏切换到蜂鸣器控制界面,如图 4-7-8 所示。



图 4-7-8 蜂鸣器控制界面

(12)点击按钮【打开】即可打开蜂鸣器,如图 4-7-9 所示。



图 4-7-9 打开蜂鸣器

(13)点击按钮【关闭】即可关闭蜂鸣器,如图 4-7-8 所示。

4.7.2 任务设计(Design)

该任务对应示例程序的设计思路如下:

- (1)客户端 App 使用 JSON 封装控制指令;
- (2)客户端 App 将控制指令通过网络发送给服务端;
- (3)服务端解析从客户端 App 发送过来的 JSON 数据;
- (4)服务端通过控制指令打开或关闭沙盘上的受控设备。

4.7.3 任务实现(Implement)

所有团队成员召开计划会议,并将会议纪要记录在表 4-7-1 中。

表 4-7-1 会议纪要记录表

5W2H	结论
WHAT	
HOW	
WHY	
WHEN	
WHERE	
WHO	
HOW MUCH	

在计划会议结束之前,所有团队成员把子任务进一步细分为更小的任务,把这些任务写在便签上,并把便签贴在看板上的“ToDo”列。

所有团队成员每天早上召开 15 分钟的站立会议,每人汇报三个问题,并更新看板上

的“ToDo”、“Doing”和“Done”三列。

本任务对应示例程序的核心代码如下：

(1)lenovo.agriculture.manualcontrol.request.ControlRequest 类中的核心代码

```
/* *
 * 执行 JSON 协议时的获取 body 函数
 * @see lenovo.agriculture.manualcontrol.request.BaseRequest#onGetJsonBody()
 */
@Override
protected String onGetJsonBody() {
    JSONObject jsonObj = new JSONObject();
    try {
        jsonObj.put("username", userName);
        // 是否要控制风扇
        if (isCtrlBlower) {
            jsonObj.put("Blower", blower);
        }
        // 是否要控制蜂鸣器
        if (isCtrlBuzzer) {
            jsonObj.put("Buzzer", buzzer);
        }
        // 是否要控制路灯
        if (isCtrlRoadLamp) {
            jsonObj.put("Roadlamp", roadLamp);
        }
        // 是否要控制水泵
        if (isCtrlWater) {
            jsonObj.put("WaterPump", waterPump);
        }
        return jsonObj.toString();
    } catch (JSONException e) {
        e.printStackTrace();
    }
    return super.onGetJsonBody();
}
```

(2)lenovo.agriculture.manualcontrol.activity.MainActivity 类中的核心代码

```
/* *
 * 设置网络连接的线程
 * @param request 请求对象
 */
private void startRequest(BaseRequest request) {
    if (request != null) {
```

```
mThread = new RequestThread(this, mApp.getHandler());
mThread.setRequest(request);
mThread.start();
}

}

/*
 * 设置请求对象
 */
private void setRequest() {
    // 设置获取控制器状态请求对象
    mGetControllerStatusRequest = new GetControllerStatusRequest("");
    mGetControllerStatusRequest.setControllerStatus(mApp
        .getControllerStatus());
    mGetControllerStatusRequest
        .setOnResponseEventListener(new OnResponseEventListener() {

            @Override
            public void onResponse(BaseRequest request,
                RequestResult result) {
                mDialog.dismiss();
                if (mGetControllerStatusRequest.isSuccess()) {
                    // 获取控制器状态以后,刷新所有控制器的开关状态
                    ControllerStatus status = mGetControllerStatusRequest
                        .getControllerStatus();
                    mBlowerCtrl.isOpen = status.getBlower();
                    mRoadlampCtrl.isOpen = status.getRoadLamp();
                    mWaterCtrl.isOpen = status.getWaterPump();
                    mBuzzerCtrl.isOpen = status.getBuzzer();
                    mAdapter.notifyDataSetChanged();
                }
            }
        });
    }

    // 设置控制器状态请求对象
    mControlRequest = new ControlRequest("");
    mControlRequest
        .setOnResponseEventListener(new OnResponseEventListener() {

            @Override
            public void onResponse(BaseRequest request,
                RequestResult result) {
                if (mControlRequest.isSuccess()) {
```

// 控制器请求成功以后,等待 1 秒钟再向服务器查询控制器状态(因为控制是异步操作)

```
RequestThread thread = new RequestThread(  
    MainActivity.this, mApp.getHandler());  
thread.setRequest(mGetControllerStatusRequest);  
try {  
    Thread.sleep(1000);  
} catch (InterruptedException e) {  
    e.printStackTrace();  
}  
thread.start();  
} else {  
    mDialog.dismiss();  
}  
}  
});  
}
```

(3)res/layout/controller_item.xml 中的核心代码

```
<? xml version="1.0" encoding="UTF-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center"  
    android:background="@android:color/transparent"  
    android:gravity="center"  
    android:orientation="vertical">
```

```
<TextView  
    android:id="@+id/name_text_view"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:gravity="center"  
    android:text="@string/buzzer"  
    android:textColor="@android:color/black"  
    android:textSize="28sp"  
    android:textStyle="bold"/>
```

```
<TextView  
    android:id="@+id/status_text_view"  
    style="@style/FontSize"  
    android:layout_width="match_parent"
```

```

    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="@string/current_status"
    android:textColor="@android:color/black"/>
<ImageView
    android:id="@+id/controller_icon"
    android:layout_width="150dp"
    android:layout_height="150dp"
    android:layout_marginBottom="10dp"
    android:layout_marginTop="10dp"
    android:contentDescription="@string/app_name"
    android:src="@drawable/buzzer_on"/>

<TextView
    android:id="@+id/open_text_view"
    style="@style/FontSize"
    android:layout_width="150dp"
    android:layout_height="50dp"
    android:layout_marginLeft="30dp"
    android:layout_marginRight="30dp"
    android:background="@drawable	btn_blue"
    android:gravity="center"
    android:text="@string/open"
    android:textColor="@android:color/white"/>

</LinearLayout>

```

(4) res/values/strings.xml 中的核心代码

```

<? xml version="1.0" encoding="utf-8"? >
<resources>

    <string name="app_name">ManualControl</string>
    <string name="water_pump">水泵</string>
    <string name="blower">风扇</string>
    <string name="road_lamp">光照</string>
    <string name="buzzer">报警</string>
    <string name="open">打开</string>
    <string name="close">关闭</string>
    <string name="current_status">当前状态:</string>
    <string name="waite">数据获取中,请等待。。。</string>
    <string name="input_ip_addr">请输入 IP 地址</string>
    <string name="invalid_ip">无效的 IP 地址</string>

```

```
<string name= "prompt">提示</string>
<string name= "no_net_work">请连接网络</string>

</resources>
(5)AndroidManifest. xml 中的核心代码
<? xml version= "1. 0" encoding= "utf-8"? >
<manifest xmlns:android= "http://schemas.android.com/apk/res/android"
    package= "lenovo.agriculture.manualcontrol"
    android:versionCode= "1"
    android:versionName= "1. 0" >

    <uses-sdk
        android:minSdkVersion= "15"
        android:targetSdkVersion= "21"/>
    <uses-permission android:name= "android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name= "android.permission.ACCESS_WIFI_STATE"/>
    <uses-permission android:name= "android.permission.INTERNET"/>

    <application
        android:name= ".ClientApp"
        android:allowBackup= "true"
        android:icon= "@drawable/ic_launcher"
        android:label= "@string/app_name"
        android:theme= "@style/AppTheme">
        <activity
            android:name= ".activity.MainActivity"
            android:label= "@string/app_name"
            android:screenOrientation= "landscape">
            <intent-filter>
                <action android:name= "android.intent.action.MAIN"/>

                <category android:name= "android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
    </application>

</manifest>
```

4. 7. 4 任务运作(Operate)

开发工程师之间两两交叉代码评审，并根据评审纪要重构代码。

第 4 章 项目实现 (Implement) —— 编码和测试

所有团队成员召开评审会，并填写表 4-7-2。

表 4-7-2

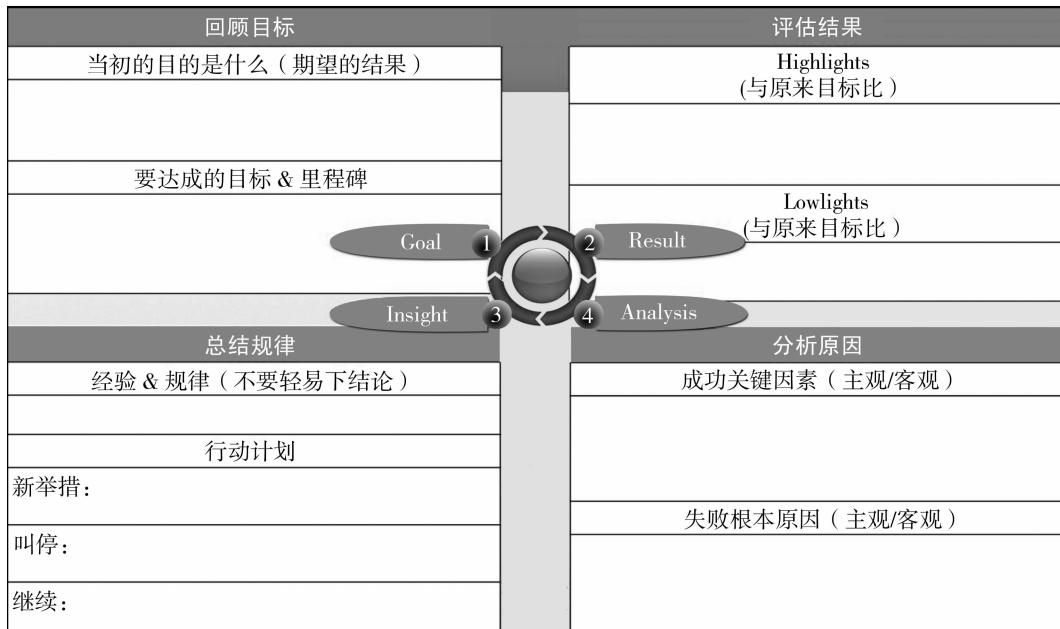
评审建议表

序号	建议描述	是否接纳	原因
1		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
2		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
3		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
4		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
5		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
6		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
7		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
8		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
9		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
10		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	

所有团队成员召开复盘会，并填写表 4-7-3。

表 4-7-3

复盘会总结表



同学们，根据塔克曼阶梯理论，你们的团队目前处于哪个阶段呢？如果还没有进入成熟阶段，那就让我们一起努力吧，在接下来的任务中，更加注重相互配合和协作，及时发现并改正团队配合过程中出现的问题，早日让我们的团队进入成熟阶段！

4.7.5 任务扩展(Extend)

课后练习：在该任务对应示例程序的基础上，使用 XML 文件替换 JSON 来封装控制指令。

子任务 8 实现系统设置界面的布局

- 任务目标：

- 实现系统设置界面的布局

- 课时分配：

- 4 课时

4.8.1 任务构思(Conceive)

绝大多数商业 App 都有系统设置模块，联想智能农业系统也不例外。本任务对应示例程序的运行效果如下：

(1) 启动示例程序，进入程序的系统设置主界面，如图 4-8-1 所示。



图 4-8-1 程序主界面

(2) 点击【语言设置】，弹出语言选择对话框，如图 4-8-2 所示。



图 4-8-2 语言选择对话框

(3) 选择【英语】，如图 4-8-3 所示。



图 4-8-3 选择【英语】

(4) 点击按钮【保存】，如图 4-8-4 所示。



图 4-8-4 点击按钮【保存】

(5)点击【自动/手动控制】，弹出语言选择对话框，如图 4-8-5 所示。



图 4-8-5 点击【自动/手动控制】

(6)选择【自动】，如图 4-8-6 所示。



图 4-8-6 选择【自动】

(7)点击按钮【保存】，如图 4-8-7 所示。



图 4-8-7 点击按钮【保存】

(8)点击【CO₂浓度阀值设置】，如图 4-8-8 所示。



图 4-8-8 点击【CO₂浓度阀值设置】

(9) 在主界面点击【光照强度阀值设置】,如图 4-8-9 所示。



图 4-8-9 点击【光照强度阀值设置】

(10) 在主界面点击【空气温湿度阀值设置】,如图 4-8-10 所示。



图 4-8-10 点击【空气温湿度阀值设置】

(11) 在主界面点击【土壤温湿度阀值设置】,如图 4-8-11 所示。



图 4-8-11 点击【土壤温湿度阀值设置】

(12)当最小值或最大值为空时,点击【保存】后出现如图 4-8-12 所示的提示。

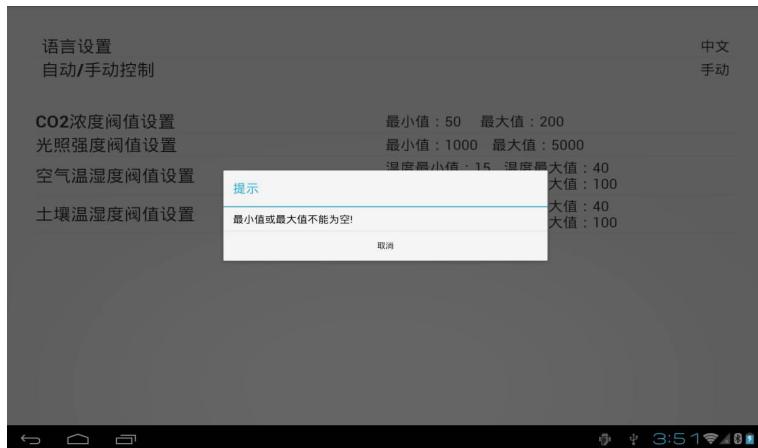


图 4-8-12 点击【保存】后出现的提示 1

(13)当最小值大于最大值时,点击【保存】后出现如图 4-8-13 所示的提示。

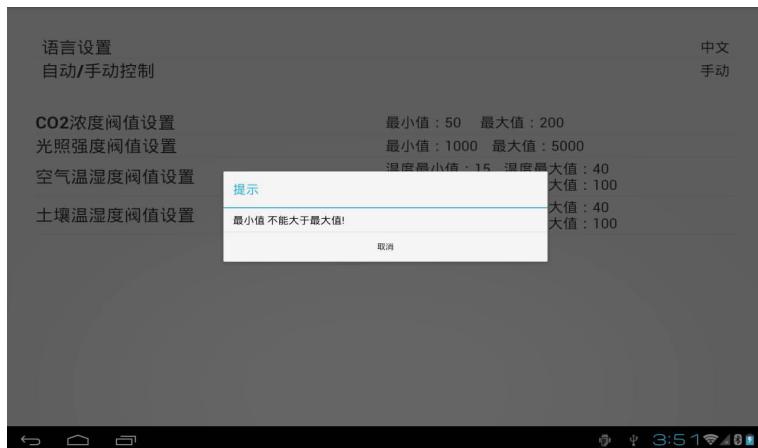


图 4-8-13 点击【保存】后出现的提示 2

4.8.2 任务设计(Design)

Android 系统提供了单独用于系统设置界面布局的标签,例如:PreferenceScreen、PreferenceCategory 和 EditTextPreference 等。此外,Android 系统还提供了单独用于加载系统设置界面对应布局文件的 API,例如:PreferenceActivity 和 PreferenceFragment 等。

4.8.3 任务实现(Implement)

所有团队成员召开计划会议,并将会议纪要记录在表 4-8-1 中。

表 4-8-1

会议纪要记录表

5W2H	结论
WHAT	
HOW	
WHY	
WHEN	
WHERE	
WHO	
HOW MUCH	

在计划会议结束之前,所有团队成员把子任务进一步细分为更小的任务,把这些任务写在便签上,并把便签贴在看板上的“ToDo”列。

所有团队成员每天早上召开 15 分钟的站立会议,每人汇报三个问题,并更新看板上的“ToDo”、“Doing”和“Done”三列。

本任务对应示例程序的核心代码如下:

(1) lenovo.agriculture.systemsetting.fragment.SettingFragment 类中的核心代码

```

/* *
 * 初始化控件
 *
 */
private void initView() {
    // 获取到控件
    mLanguageETP = (EditTextPreference) findPreference(LANGUAGE);
    mAutoETP = (EditTextPreference) findPreference(AUTO);
    mCO2ETP = (EditTextPreference) findPreference(CO2);
    mLightETP = (EditTextPreference) findPreference(LIGHT);
    mAireTP = (EditTextPreference) findPreference(AIR);
    mSoilETP = (EditTextPreference) findPreference(SOIL);
    saveLanguage();
    saveAuto();
    // 设定初始值
}

```

```
mCO2ETP.setSummary(getString(R.string.min_value) + getIntValue(CO2_MIN)
+ "" + getString(R.string.max_value) + getIntValue(CO2_MAX));
mLightETP.setSummary(getString(R.string.min_value)
+ getIntValue(LIGHT_MIN) + "" + getString(R.string.max_value)
+ getIntValue(LIGHT_MAX));
mAireTP.setSummary(getString(R.string.temp_min_value)
+ getIntValue(AIR_T_MIN) + ""
+ getString(R.string.temp_max_value) + getIntValue(AIR_T_MAX)
+ '\n' + getString(R.string.humi_min_value)
+ getIntValue(AIR_H_MIN) + ""
+ getString(R.string.humi_max_value) + getIntValue(AIR_H_MAX));
mSoilETP.setSummary(getString(R.string.temp_min_value)
+ getIntValue(SOIL_T_MIN) + ""
+ getString(R.string.temp_max_value) + getIntValue(SOIL_T_MAX)
+ '\n' + getString(R.string.humi_min_value)
+ getIntValue(SOIL_H_MIN) + ""
+ getString(R.string.humi_max_value) + getIntValue(SOIL_H_MAX));
}
```

(2)res/layout/one_item.xml 中的核心代码

```
<? xml version="1.0" encoding="utf-8"? >
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:baselineAligned="false"
    android:orientation="horizontal">

    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_margin="30dp"
        android:layout_weight="1"
        android:gravity="center"
        android:orientation="horizontal">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/min_value"
            android:textColor="@android:color/black"
            android:textSize="25sp"/>
```

```
<EditText
    android:id="@+id/major_min_value_edit_text"
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:inputType="numberSigned"
    android:maxLength="5"
    android:singleLine="true"
    android:textSize="25sp"/>
</LinearLayout>

<LinearLayout
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_margin="30dp"
    android:layout_weight="1"
    android:gravity="center"
    android:orientation="horizontal">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/max_value"
        android:textColor="@android:color/black"
        android:textSize="25sp"/>

    <EditText
        android:id="@+id/major_max_value_edit_text"
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:inputType="numberSigned"
        android:maxLength="5"
        android:singleLine="true"
        android:textSize="25sp"/>
</LinearLayout>
```

</LinearLayout>

(3) res/layout/temp_humi.xml 中的核心代码

```
<? xml version="1.0" encoding="utf-8"? >
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
```

```
    android:gravity= "center"
    android:orientation= "vertical">

<LinearLayout
    android:layout_width= "match_parent"
    android:layout_height= "0dp"
    android:layout_weight= "1"
    android:baselineAligned= "false"
    android:gravity= "center"
    android:orientation= "vertical">

    <TextView
        android:layout_width= "wrap_content"
        android:layout_height= "wrap_content"
        android:text= "@string/temp_threshold_value_setting"
        android:textColor= "@android:color/black"
        android:textSize= "25sp"/>

    <LinearLayout
        android:layout_width= "match_parent"
        android:layout_height= "wrap_content"
        android:layout_margin= "30dp"
        android:baselineAligned= "false"
        android:gravity= "center"
        android:orientation= "horizontal">

        <LinearLayout
            android:layout_width= "0dp"
            android:layout_height= "wrap_content"
            android:layout_weight= "1"
            android:gravity= "center"
            android:orientation= "horizontal">

            <TextView
                android:layout_width= "wrap_content"
                android:layout_height= "wrap_content"
                android:text= "@string/min_value"
                android:textColor= "@android:color/black"
                android:textSize= "25sp"/>

            <EditText
```

```
        android:id="@+id/major_min_value_edit_text"
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:inputType="numberSigned"
        android:maxLength="5"
        android:singleLine="true"
        android:textSize="25sp"/>
    </LinearLayout>

    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:gravity="center"
        android:orientation="horizontal">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="@string/max_value"
            android:textColor="@android:color/black"
            android:textSize="25sp"/>

        <EditText
            android:id="@+id/major_max_value_edit_text"
            android:layout_width="100dp"
            android:layout_height="wrap_content"
            android:inputType="numberSigned"
            android:maxLength="5"
            android:singleLine="true"
            android:textSize="25sp"/>
    </LinearLayout>
</LinearLayout>
</LinearLayout>

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_weight="1"
    android:baselineAligned="false"
    android:gravity="center"
```

```
    android:orientation= "vertical">

    <TextView
        android:layout_width= "wrap_content"
        android:layout_height= "wrap_content"
        android:text= "@string/humi_threshold_value_setting"
        android:textColor= "@android:color/black"
        android:textSize= "25sp"/>

    <LinearLayout
        android:layout_width= "match_parent"
        android:layout_height= "wrap_content"
        android:layout_margin= "30dp"
        android:baselineAligned= "false"
        android:gravity= "center"
        android:orientation= "horizontal">

        <LinearLayout
            android:layout_width= "0dp"
            android:layout_height= "wrap_content"
            android:layout_weight= "1"
            android:gravity= "center"
            android:orientation= "horizontal">

            <TextView
                android:layout_width= "wrap_content"
                android:layout_height= "wrap_content"
                android:text= "@string/min_value"
                android:textColor= "@android:color/black"
                android:textSize= "25sp"/>

            <EditText
                android:id= "@+id/slave_min_value_edit_text"
                android:layout_width= "100dp"
                android:layout_height= "wrap_content"
                android:inputType= "numberSigned"
                android:maxLength= "5"
                android:singleLine= "true"
                android:textSize= "25sp"/>
        </LinearLayout>
    </LinearLayout>
```

```

<LinearLayout
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:gravity="center"
    android:orientation="horizontal">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/max_value"
        android:textColor="@android:color/black"
        android:textSize="25sp"/>

    <EditText
        android:id="@+id/slave_max_value_edit_text"
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:inputType="numberSigned"
        android:maxLength="5"
        android:singleLine="true"
        android:textSize="25sp"/>

```

</LinearLayout>

</LinearLayout>

</LinearLayout>

</LinearLayout>

(4) res/layout/radio_item.xml 中的核心代码

```

<? xml version="1.0" encoding="utf-8"? >
<RadioGroup xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/radioGroup1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center">

    <RadioButton
        android:id="@+id/radio0"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:checked="true"
        android:textSize="25sp"/>

```

```
<RadioButton
    android:id="@+id/radio1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:textSize="25sp"/>

</RadioGroup>
(5)res/layout/linear_item.xml 中的核心代码
<? xml version="1.0" encoding="utf-8"? >
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <TextView
        android:id="@+android:id/title"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:gravity="center_vertical"
        android:textSize="30sp"
        android:textStyle="bold"/>

    <TextView
        android:id="@+android:id/summary"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_gravity="end"
        android:layout_weight="1"
        android:gravity="center_vertical"
        android:textSize="25sp"/>

</LinearLayout>
(6)res/layout/relative_item.xml 中的核心代码
<? xml version="1.0" encoding="utf-8"? >
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">
```

```
<TextView
    android:id="@+android:id/title"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:gravity="center_vertical"
    android:paddingLeft="10dp"
    android:textSize="30sp"
    android:textStyle="bold"/>

<TextView
    android:id="@+android:id/summary"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentRight="true"
    android:layout_centerVertical="true"
    android:layout_marginRight="10dp"
    android:gravity="center_vertical"
    android:textSize="25sp"/>

</RelativeLayout>
(7)res/values/strings.xml中的核心代码
<? xml version="1.0" encoding="utf-8"? >
<resources>

    <string name="app_name">SystemSetting</string>
    <string name="language_setting">语言设置</string>
    <string name="chinese">中文</string>
    <string name="english">英语</string>
    <string name="auto_control">自动/手动控制</string>
    <string name="auto">自动</string>
    <string name="manual">手动</string>
    <string name="co2_setting">CO2 浓度阀值设置</string>
    <string name="light_setting">光照强度阀值设置</string>
    <string name="air_setting">空气温湿度阀值设置</string>
    <string name="soil_setting">土壤温湿度阀值设置</string>
    <string name="temp_threshold_value_setting">温度阀值设置</string>
    <string name="humi_threshold_value_setting">湿度阀值设置</string>
    <string name="min_value">最小值:</string>
    <string name="max_value">最大值:</string>
    <string name="temp_min_value">温度最小值:</string>
    <string name="temp_max_value">温度最大值:</string>
```

```
<string name= "humi_min_value">湿度最小值:</string>
<string name= "humi_max_value">湿度最大值:</string>
<string name= "min_max_value_not_empty">最小值或最大值不能为空! </string>
<string name= "min_more_than_max">最小值不能大于最大值! </string>
<string name= "prompt">提示</string>
<string name= "cancel">取消</string>
<string name= "ok">保存</string>

</resources>
(8)res/xml/preference_setting.xml 中的核心代码
<? xml version= "1.0" encoding= "utf-8"? >
<PreferenceScreen xmlns:android= "http://schemas.android.com/apk/res/android">

    <PreferenceCategory android:layout= "@layout/category_item">
        <EditTextPreference
            android:dialogLayout= "@layout/radio_item"
            android:dialogTitle= "@string/language_setting"
            android:key= "language"
            android:layout= "@layout/relative_item"
            android:negativeButtonText= "@string/cancel"
            android:positiveButtonText= "@string/ok"
            android:summary= "@string/chinese"
            android:title= "@string/language_setting"/>
        <EditTextPreference
            android:dialogLayout= "@layout/radio_item"
            android:dialogTitle= "@string/auto_control"
            android:key= "auto"
            android:layout= "@layout/relative_item"
            android:negativeButtonText= "@string/cancel"
            android:positiveButtonText= "@string/ok"
            android:summary= "@string/manual"
            android:title= "@string/auto_control"/>
    </PreferenceCategory>
    <PreferenceCategory android:layout= "@layout/category_item">
        <EditTextPreference
            android:dialogLayout= "@layout/one_item"
            android:dialogTitle= "@string/co2_setting"
            android:key= "co2"
            android:layout= "@layout/linear_item"
            android:negativeButtonText= "@string/cancel"
            android:positiveButtonText= "@string/ok"
        </EditTextPreference>
    </PreferenceCategory>
</PreferenceScreen>
```

```

        android:title="@string/co2_setting"/>
<EditTextPreference
        android:dialogLayout="@layout/one_item"
        android:DialogTitle="@string/light_setting"
        android:key="light"
        android:layout="@layout/linear_item"
        android:negativeButtonText="@string/cancel"
        android:positiveButtonText="@string/ok"
        android:title="@string/light_setting"/>
<EditTextPreference
        android:dialogLayout="@layout/temp_humi"
        android:DialogTitle="@string/air_setting"
        android:key="air"
        android:layout="@layout/linear_item"
        android:negativeButtonText="@string/cancel"
        android:positiveButtonText="@string/ok"
        android:title="@string/air_setting"/>
<EditTextPreference
        android:dialogLayout="@layout/temp_humi"
        android:DialogTitle="@string/soil_setting"
        android:key="soil"
        android:layout="@layout/linear_item"
        android:negativeButtonText="@string/cancel"
        android:positiveButtonText="@string/ok"
        android:title="@string/soil_setting"/>
</PreferenceCategory>

</PreferenceScreen>

```

(9) AndroidManifest.xml 中的核心代码

```

<? xml version="1.0" encoding="utf-8"? >
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="lenovo.agriculture.systemsetting"
    android:versionCode="1"
    android:versionName="1.0">

    <uses-sdk
        android:minSdkVersion="15"
        android:targetSdkVersion="21"/>

    <application
        android:allowBackup="true"

```

```

    android:icon= "@drawable/ic_launcher"
    android:label= "@string/app_name"
    android:theme= "@style/AppTheme">
<activity
    android:name= ".activity.MainActivity"
    android:label= "@string/app_name">
    <intent-filter>
        <action android:name= "android.intent.action.MAIN"/>

        <category android:name= "android.intent.category.LAUNCHER"/>
    </intent-filter>
</activity>
</application>

</manifest>

```

4.8.4 任务运作(Operate)

开发工程师之间两两交叉代码评审，并根据评审纪要重构代码。

所有团队成员召开评审会，并填写表 4-8-2。

表 4-8-2

评审建议表

序号	建议描述	是否接纳	原因
1		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
2		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
3		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
4		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
5		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
6		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
7		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
8		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
9		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
10		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	

所有团队成员召开复盘会，并填写表 4-8-3。

表 4-8-3

复盘会总结表

回顾目标		评估结果	
当初的目的是什么(期望的结果)		Highlights (与原来目标比)	
要达成的目标&里程碑		Lowlights (与原来目标比)	
Goal	1	Result	2
Insight	3	Analysis	4
总结规律		分析原因	
经验&规律(不要轻易下结论)		成功关键因素(主观/客观)	
行动计划		失败根本原因(主观/客观)	
新举措:			
叫停:			
继续:			

同学们,根据塔克曼阶梯理论,你们的团队目前处于哪个阶段呢?如果还没有进入成熟阶段,那就让我们一起努力吧,在接下来的任务中,更加注重相互配合和协作,及时发现并改正团队配合过程中出现的问题,早日让我们的团队进入成熟阶段!

4.8.5 任务扩展(Extend)

课后练习:在该任务对应示例程序的基础上,在设置界面添加一个 CheckBox。

子任务9 实现应用程序的国际化

- 任务目标:

 - 实现应用程序的国际化

- 课时分配:

 - 6 课时

4.9.1 任务构思(Conceive)

在联想智能农业系统中,为了能支持中文和英文两种语言,需要实现应用程序的国际化。

本任务对应示例程序的运行效果如下:

(1)启动示例程序,进入程序的系统设置主界面,如图 4-9-1 所示。

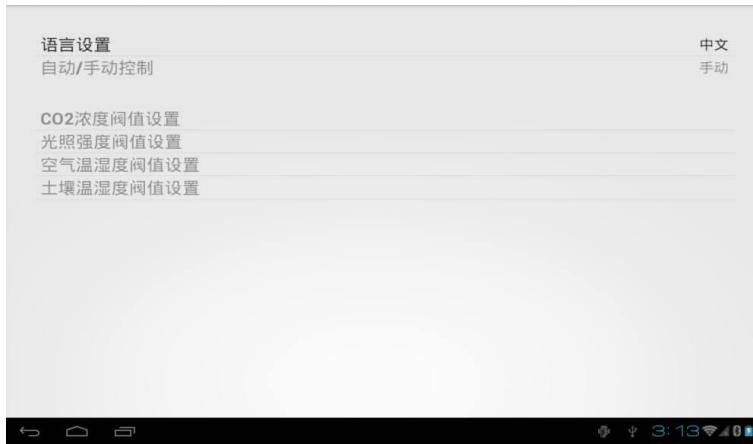


图 4-9-1 程序主界面

(2)点击【语言设置】,界面会弹出语言选择的对话框,如图 4-9-2 所示。



图 4-9-2 语言选择对话框

(3)选择【英语】选项,如图 4-9-3 所示。



图 4-9-3 选择【英语】选项

(4) 点击按钮【保存】，切换到英语，如图 4-9-4 所示。

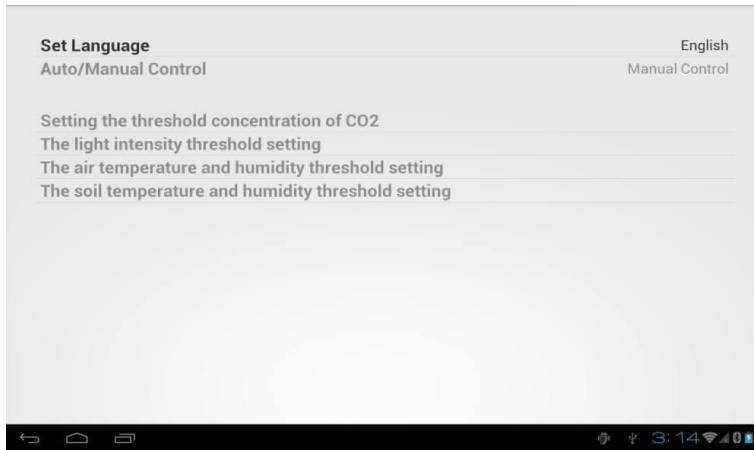


图 4-9-4 切换到英语界面

4.9.2 任务设计(Design)

本任务对应示例程序的设计思路是：

(1) 在 res 目录创建对应不同国家的 values-xx 文件夹，中文对应的文件夹名称是 values-zh，英文对应的文件夹名称是 values-en。

(2) 当应用程序运行时，系统会根据当前的语言设置自动到对应的 values-xx 文件夹中加载资源。

4.9.3 任务实现(Implement)

所有团队成员召开计划会议，并将会议纪要记录在表 4-9-1 中。

表 4-9-1 会议纪要记录表

5W2H	结论
WHAT	
HOW	
WHY	
WHEN	
WHERE	
WHO	
HOW MUCH	

在计划会议结束之前，所有团队成员把子任务进一步细分为更小的任务，把这些任务写在便签上，并把便签贴在看板上的“ToDo”列。

所有团队成员每天早上召开 15 分钟的站立会议，每人汇报三个问题，并更新看板上的“ToDo”、“Doing”和“Done”三列。

本任务对应示例程序的核心代码如下：

(1) lenovo.agriculture.changelanguage.fragment.SettingFragment 类中的核心代码

```
/* *
 * 切换语言
 * @param locale 本地语言的参数
 */
private void changeLanguage(Locale locale) {
    Resources res = getResources();
    Configuration config = res.getConfiguration();
    config.locale = locale;
    DisplayMetrics dm = res.getDisplayMetrics();
    res.updateConfiguration(config, dm);
    getActivity().sendBroadcast(new Intent("language"));
}
```

(2) lenovo.agriculture.changelanguage.Receiver.ChangeReceiver 类中的核心代码

```
mIntent = new Intent(context, MainActivity.class);
mIntent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
context.startActivity(mIntent);
```

(3) res/values/strings.xml 中的核心代码

```
<? xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">语言切换</string>
    <string name="language_setting">语言设置</string>
    <string name="chinese">中文</string>
    <string name="english">英语</string>
    <string name="auto_control">自动/手动控制</string>
    <string name="manual">手动</string>
    <string name="co2_setting">CO2 浓度阀值设置</string>
    <string name="light_setting">光照强度阀值设置</string>
    <string name="air_setting">空气温湿度阀值设置</string>
    <string name="soil_setting">土壤温湿度阀值设置</string>
    <string name="cancel">取消</string>
    <string name="ok">保存</string>

</resources>
```

(4) res/values-en/strings.xml 中的核心代码

```
<? xml version="1.0" encoding="utf-8"?>
<resources>
```

```

<string name= "app_name">ChangeLanguage</string>
<string name= "language_setting">Set Language</string>
<string name= "chinese">Chinese</string>
<string name= "english">English</string>
<string name= "auto_control">Auto/Manual Control</string>
<string name= "manual">Manual Control</string>
<string name= "co2_setting">Setting the threshold concentration of CO2</string>
<string name= "light_setting">The light intensity threshold setting</string>
<string name= "air_setting">The air temperature and humidity threshold
setting</string>
<string name= "soil_setting">The soil temperature and humidity threshold
setting</string>
<string name= "cancel">Cancel</string>
<string name= "ok">Save</string>

</resources>

```

(5) res/xml/preference_setting.xml 中的核心代码

```

<? xml version= "1.0" encoding= "utf-8"? >
<PreferenceScreen xmlns:android= "http://schemas.android.com/apk/res/android">

    <PreferenceCategory android:layout= "@layout/category_item">
        <EditTextPreference
            android:dialogLayout= "@layout/radio_item"
            android:dialogTitle= "@string/language_setting"
            android:key= "language"
            android:layout= "@layout/relative_item"
            android:negativeButtonText= "@string/cancel"
            android:positiveButtonText= "@string/ok"
            android:title= "@string/language_setting"/>
        <EditTextPreference
            android:enabled= "false"
            android:layout= "@layout/relative_item"
            android:summary= "@string/manual"
            android:title= "@string/auto_control"/>
    </PreferenceCategory>
    <PreferenceCategory android:layout= "@layout/category_item">
        <EditTextPreference
            android:enabled= "false"
            android:layout= "@layout/relative_item"
            android:title= "@string/co2_setting"/>
        <EditTextPreference
            android:enabled= "false"

```

```
    android:layout="@layout/relative_item"
    android:title="@string/light_setting"/>
<EditTextPreference
    android:enabled="false"
    android:layout="@layout/relative_item"
    android:title="@string/air_setting"/>
<EditTextPreference
    android:enabled="false"
    android:layout="@layout/relative_item"
    android:title="@string/soil_setting"/>
</PreferenceCategory>
```

```
</PreferenceScreen>
```

(6) AndroidManifest.xml 中的核心代码

```
<? xml version="1.0" encoding="utf-8"? >
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="lenovo.agriculture.changelanguage"
    android:versionCode="1"
    android:versionName="1.0">

    <uses-sdk
        android:minSdkVersion="15"
        android:targetSdkVersion="21"/>

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme">
        <activity
            android:name=".activity.MainActivity"
            android:label="@string/app_name"
            android:screenOrientation="landscape">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>

                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>

        <receiver android:name="lenovo.agriculture.changelanguage.Receiver.ChangeReceiver">
            <intent-filter>
                <action android:name="language"/>
            </intent-filter>
        </receiver>
    </application>
</manifest>
```

```

</intent-filter>
</receiver>
</application>

</manifest>

```

4.9.4 任务运作(Operate)

开发工程师之间两两交叉代码评审，并根据评审纪要重构代码。

所有团队成员召开评审会，并填写表 4-9-2。

表 4-9-2

评审建议表

序号	建议描述	是否接纳	原因
1		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
2		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
3		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
4		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
5		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
6		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
7		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
8		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
9		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
10		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	

所有团队成员召开复盘会，并填写表 4-9-3。

表 4-9-3

复盘会总结表

The diagram illustrates the flow of the post-mortem process. It starts with 'Goal' (1), leading to 'Result' (2). From 'Result', it branches into 'Insight' (3) and 'Analysis' (4). Finally, 'Analysis' leads back to 'Goal'.

回顾目标		评估结果	
当初的目的是什么 (期望的结果)		Highlights (与原来目标比)	
要达成的目标 & 里程碑		Lowlights (与原来目标比)	
Goal 1		Result 2	
Insight 3		Analysis 4	
总结规律		分析原因	
经验 & 规律 (不要轻易下结论)		成功关键因素 (主观/客观)	
行动计划		失败根本原因 (主观/客观)	
新举措:			
叫停:			
继续:			

同学们,根据塔克曼阶梯理论,你们的团队目前处于哪个阶段呢?如果还没有进入成熟阶段,那就让我们一起努力吧,在接下来的任务中,更加注重相互配合和协作,及时发现并改正团队配合过程中出现的问题,早日让我们的团队进入成熟阶段!

4.9.5 任务扩展(Extend)

课后练习:在该任务对应示例程序的基础上,添加对繁体中文的支持。

子任务 10 自动打开或关闭沙盘上的受控设备

- 任务目标:

- 自动打开或关闭沙盘上的受控设备

- 课时分配:

- 4 课时

4.10.1 任务构思(Conceive)

在联想智能农业系统中,客户端 App 除了可以手动打开或关闭沙盘上的受控设备,还可以根据用户设置的各个传感器的阀值范围自动打开或关闭沙盘上的受控设备。

本任务对应示例程序的运行效果如下:

(1)启动服务端 App,主界面如图 4-10-1 所示。



图 4-10-1 程序主界面

(2)将【服务状态】设置为打开,如图 4-10-2 所示。



图 4-10-2 【服务状态】设置为打开

(3) 启动客户端 App, 设置要连接的服务端 App 的 IP 地址, 如图 4-10-3 所示。



图 4-10-3 设置 IP 地址

(4) 点击按钮【确定】, 进入系统设置主界面, 如图 4-10-4 所示。



图 4-10-4 系统设置主界面

(5) 点击【自动/手动控制】，界面会弹出控制方式对话框, 如图 4-10-5 所示。



图 4-10-5 控制方式对话框

(6)选择【自动】选项,如图 4-10-6 所示。



图 4-10-6 选择【自动】选项

(7)点击按钮【保存】,将控制方式切换到自动控制,界面如图 4-10-7 所示。

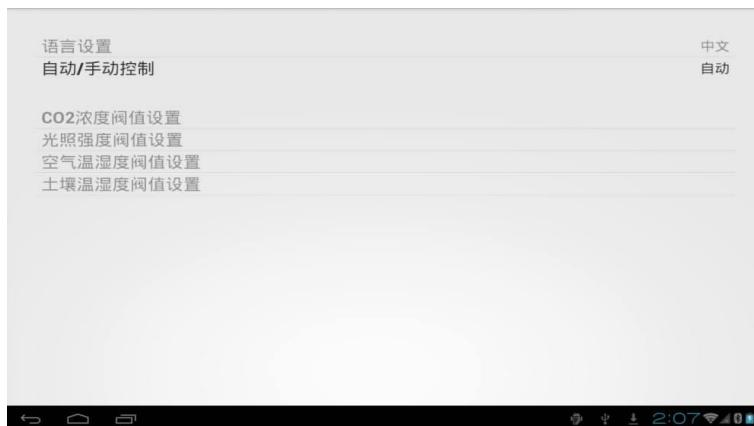


图 4-10-7 控制方式切换到自动控制

(8)如步骤(6)~(7)所示,在弹出的控制方式对话框选择手动选项,就可将控制方式切换到手动控制。

4.10.2 任务设计(Design)

本任务对应示例程序的主要设计思路是：

(1) 客户端 App 设置各个传感器的阀值范围，将其保存在服务端；

(2) 服务端每隔 1 秒都会判断当前某个传感器的数值是否超出了其阀值范围。如果超出了阀值范围，就会根据业务逻辑自动打开或关闭沙盘上的受控设备。

4.10.3 任务实现(Implement)

所有团队成员召开计划会议，并将会议纪要记录在表 4-10-1 中。

表 4-10-1

会议纪要记录表

5W2H	结论
WHAT	
HOW	
WHY	
WHEN	
WHERE	
WHO	
HOW MUCH	

在计划会议结束之前，所有团队成员把子任务进一步细分为更小的任务，把这些任务写在便签上，并把便签贴在看板上的“ToDo”列。

所有团队成员每天早上召开 15 分钟的站立会议，每人汇报三个问题，并更新看板上的“ToDo”、“Doing”和“Done”三列。

本任务对应示例程序的核心代码如下：

(1) 客户端 lenovo.agriculture.automaticcontrolclient.fragment.SettingFragment 类中的核心代码

```

/* *
 * 切换手动/自动控制
 *
 * @param newValue 更新后的 Preference
 */
private void autonControl() {
    // 保存数据
    mAutoControlETP.getEditor().putInt(AUTOGROUP, mAutoId).commit();
    if (mAutoId == R.id.radio0) {
        mCurrentStat = getString(R.string.manual);
        mRequest.setAutoControl(0);
    } else {
        mCurrentStat = getString(R.string.auto);
    }
}

```

```
mRequest.setAutoControl(1);
}
startRequest(mRequest);
// 设置 Summary 的内容
mAutoControlETP.setSummary(mCurrentStat);
}
```

(2) 服务器端 lenovo.agriculture.automaticcontrolserver.serial.SerialportService 类中的核心代码

```
/*
 * 自动控制逻辑要在服务开启 10 秒以后再启动
 */
private void autoControl() {
    if (mStartTime != null && mApp.getControlAuto() > 0) {
        Date now = new Date();
        if (now.getTime() - mStartTime.getTime() > 10000) {
            Sensor pm25 = mSensorList.get(0); // PM2.5 浓度传感器
            Sensor co2 = mSensorList.get(1); // CO2 浓度传感器
            Sensor airT = mSensorList.get(2); // 空气温度传感器
            Sensor airH = mSensorList.get(3); // 空气湿度传感器
            Sensor soilT = mSensorList.get(4); // 土壤温度传感器
            Sensor soilH = mSensorList.get(5); // 土壤湿度传感器
            Sensor light = mSensorList.get(6); // 光照强度传感器

            Controller waterPump = mControllerList.get(0); // 水泵控制器
            Controller blower = mControllerList.get(1); // 风扇控制器
            Controller buzzer = mControllerList.get(2); // 蜂鸣器控制器
            Controller roadlamp = mControllerList.get(3); // 路灯控制器
            controlBuzzer(pm25, co2, airT, airH, soilT, soilH, light,
                           buzzer);
            controlWaterPump(soilH, waterPump);
            controlBlower(co2, blower);
            controlLight(light, roadlamp);
        }
    }
}
```

(3) 客户端 res/values/strings.xml 中的核心代码

```
<? xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">AutomaticControlClient</string>
    <string name="language_setting">语言设置</string>
    <string name="chinese">中文</string>
```

```

<string name= "auto_control">自动/手动控制</string>
<string name= "auto">自动</string>
<string name= "manual">手动</string>
<string name= "co2_setting">CO2 浓度阀值设置</string>
<string name= "light_setting">光照强度阀值设置</string>
<string name= "air_setting">空气温湿度阀值设置</string>
<string name= "soil_setting">土壤温湿度阀值设置</string>
<string name= "prompt">提示</string>
<string name= "input_ip_addr">请输入 IP 地址</string>
<string name= "invalid_ip">无效的 IP 地址</string>
<string name= "no_net_work">请连接网络</string>
<string name= "cancel">取消</string>
<string name= "ok">保存</string>

```

</resources>

(4) 客户端 res/xml/preference_setting.xml 中的核心代码

```

<? xml version= "1.0" encoding= "utf-8"? >
<PreferenceScreen xmlns:android= "http://schemas.android.com/apk/res/android">

    <PreferenceCategory android:layout= "@layout/category_item">
        <EditTextPreference
            android:enabled= "false"
            android:layout= "@layout/relative_item"
            android:summary= "@string/chinese"
            android:title= "@string/language_setting"/>
        <EditTextPreference
            android:dialogLayout= "@layout/radio_item"
            android:DialogTitle= "@string/auto_control"
            android:key= "auto"
            android:layout= "@layout/relative_item"
            android:negativeButtonText= "@string/cancel"
            android:positiveButtonText= "@string/ok"
            android:summary= "@string/manual"
            android:title= "@string/auto_control"/>
    </PreferenceCategory>
    <PreferenceCategory android:layout= "@layout/category_item">
        <EditTextPreference
            android:enabled= "false"
            android:layout= "@layout/relative_item"
            android:title= "@string/co2_setting"/>
        <EditTextPreference
            android:enabled= "false"

```

```
        android:layout="@layout/relative_item"
        android:title="@string/light_setting"/>
    <EditTextPreference
        android:enabled="false"
        android:layout="@layout/relative_item"
        android:title="@string/air_setting"/>
    <EditTextPreference
        android:enabled="false"
        android:layout="@layout/relative_item"
        android:title="@string/soil_setting"/>
</PreferenceCategory>

</PreferenceScreen>
(5)客户端 AndroidManifest.xml 中的核心代码
<? xml version="1.0" encoding="utf-8"? >
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="lenovo.agriculture.automaticcontrolclient"
    android:versionCode="1"
    android:versionName="1.0">

    <uses-sdk
        android:minSdkVersion="15"
        android:targetSdkVersion="21"/>

    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name="android.permission.INTERNET"/>

    <application
        android:name=".ClientApp"
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme">
        <activity
            android:name=".activity.MainActivity"
            android:label="@string/app_name"
            android:screenOrientation="landscape"
            android:windowSoftInputMode="stateHidden|adjustUnspecified">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
```

```

<category android:name="android.intent.category.LAUNCHER"/>
</intent-filter>
</activity>
</application>

</manifest>

```

4.10.4 任务运作(Operate)

开发工程师之间两两交叉代码评审，并根据评审纪要重构代码。

所有团队成员召开评审会，并填写表 4-10-2。

表 4-10-2 评审建议表

序号	建议描述	是否接纳	原因
1		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
2		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
3		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
4		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
5		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
6		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
7		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
8		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
9		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
10		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	

所有团队成员召开复盘会，并填写表 4-10-3。

表 4-10-3 复盘会总结表

回顾目标		评估结果	
当初的目的是什么 (期望的结果)		Highlights (与原来目标比)	
要达成的目标 & 里程碑		Lowlights (与原来目标比)	
Goal	1	Result	2
Insight	3	Analysis	4
总结规律			
经验 & 规律 (不要轻易下结论)		分析原因	
行动计划		成功关键因素 (主观/客观)	
新举措:		失败根本原因 (主观/客观)	
叫停:			
继续:			

同学们,根据塔克曼阶梯理论,你们的团队目前处于哪个阶段呢?如果还没有进入成熟阶段,那就让我们一起努力吧,在接下来的任务中,更加注重相互配合和协作,及时发现并改正团队配合过程中出现的问题,早日让我们的团队进入成熟阶段!

4.10.5 任务扩展(Extend)

课后练习:无。

子任务 11 使用第三方控件 AChartEngine 绘制图形

- 任务目标:

- 使用第三方控件 AChartEngine 绘制图形

- 课时分配:

- 4 课时

4.11.1 任务构思(Conceive)

在联想智能农业系统中,沙盘上各个传感器的数据都是在不断动态变化的,为了将各个传感器的实时数据展示在客户端 App 以便用户方便的查看,可以通过折线图进行展示。

本任务对应示例程序的运行效果如下:

(1)启动示例程序,折线图的动态变化效果分别如图 4-11-1、图 4-11-2 和图 4-11-3 所示。

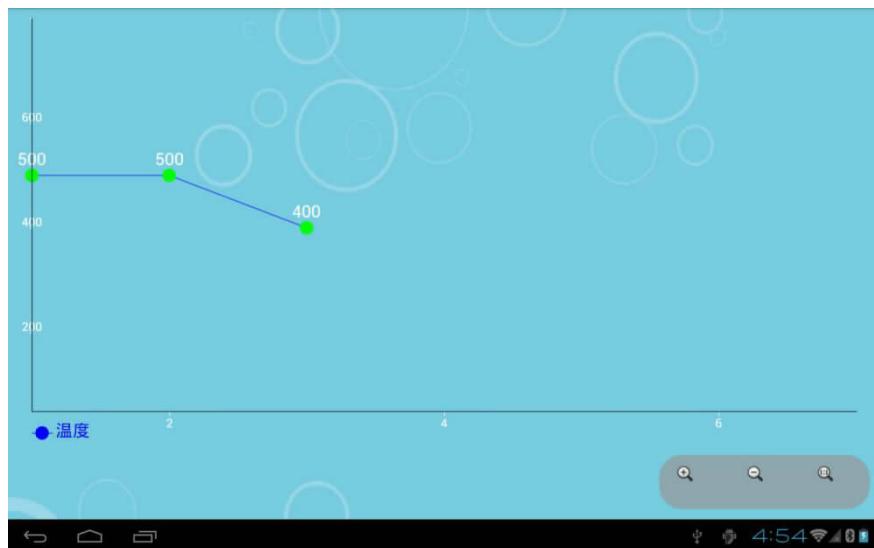


图 4-11-1 折线图的动态变化效果 1

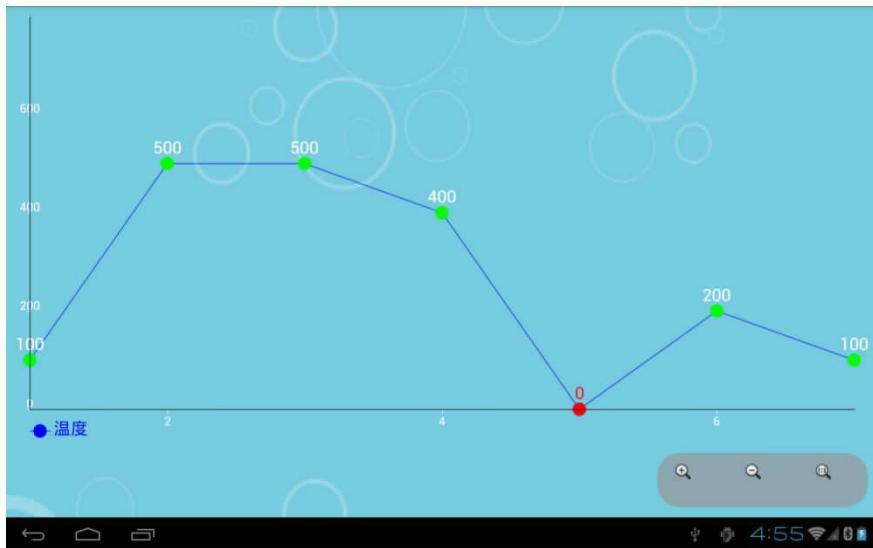


图 4-11-2 折线图的动态变化效果 2

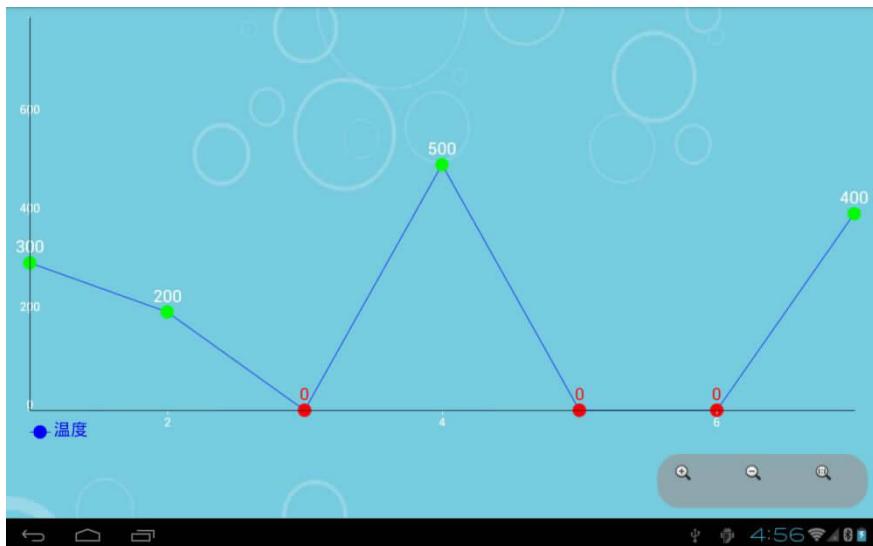


图 4-11-3 折线图的动态变化效果 3

4.11.2 任务设计(Design)

AChartEngine 是 Google 开发的用于绘制各种图形的第三方库,其使用步骤可以分为如下四步:

- (1) 创建数据集,把要显示的数据存储在数组或列表中;
- (2) 构建渲染器,把每种图形对应的渲染器构建好;
- (3) 设置渲染器,设置步骤(2)构建的渲染器的各种属性;
- (4) 将数据集和渲染器作为参数传递给 ChartFactory 的相关方法。

4.11.3 任务实现(Implement)

所有团队成员召开计划会议，并将会议纪要记录在表 4-11-1 中。

表 4-11-1

会议纪要记录表

5W2H	结论
WHAT	
HOW	
WHY	
WHEN	
WHERE	
WHO	
HOW MUCH	

在计划会议结束之前，所有团队成员把子任务进一步细分为更小的任务，把这些任务写在便签上，并把便签贴在看板上的“ToDo”列。

所有团队成员每天早上召开 15 分钟的站立会议，每人汇报三个问题，并更新看板上的“ToDo”、“Doing”和“Done”三列。

本任务对应示例程序的核心代码如下：

(1) lenovo.agriculture.chartshow.util.ChartView 类中的核心代码

```
/*
 * 绘制图表方法
 * @param context 上下文对象
 * @param chartPagerBean 图表容器
 * @param layout 展示图表的布局
 * @param xAxisMax 最大 X 轴的值
 */
public void draw(Context context, ChartPagerBean chartPagerBean,
    LinearLayout layout, int xAxisMax) {

    // 曲线图表信息对象,最多存储两个曲线图的信息
    ChartPagerBean mChartPagerBean = chartPagerBean;
    // 图表数据集合
    XYMultipleSeriesDataset mDataset;
    // 图表数据渲染器
    XYMultipleSeriesRenderer mRenderer;
    // 图表视图
    GraphicalView mChartView;
    // 创建图表数据集合
    mDataset = new XYMultipleSeriesDataset();
```

```
mRenderer = setRenderer();

int yAxisMin = mChartPagerBean.majorMin;// Y 轴的最小值
int yAxisMax = mChartPagerBean.majorMax;// Y 轴的最大值

// 创建一个数据子集
XYSeries seriesMajor = new XYSeries(mChartPagerBean.majorName);
int i = 1;
// 将数据添加到数据子集中，并动态修改最大值和最小值
for (int value : mChartPagerBean.majorValueList) {
    seriesMajor.add(i, value);
    if (value < yAxisMin) {
        yAxisMin = value;
    }
    if (value > yAxisMax) {
        yAxisMax = value;
    }
    i++;
}
// 将数据子集添加到图表数据集合中
mDataset.addSeries(seriesMajor);
ColoursXYSeriesRenderer rendererMajor = setMajorSeriesRenderer(mChartPagerBean);
// 将曲线渲染对象添加到图表渲染器中
mRenderer.addSeriesRenderer(rendererMajor);

// 如果图表中存在两条曲线，则再分别创建一个数据子集和一个曲线渲染对象
if (mChartPagerBean.isHasSlave) {
    // 创建一个数据子集
    XYSeries seriesSlave = new XYSeries(mChartPagerBean.slaveName);
    i = 1;
    // 将数据添加到数据子集中，并动态修改最大值和最小值
    for (int value : mChartPagerBean.slaveValueList) {
        value = mChartPagerBean.slaveValueList.get(i - 1);
        seriesSlave.add(i, value);
        if (value < yAxisMin) {
            yAxisMin = value;
        }
        if (value > yAxisMax) {
            yAxisMax = value;
        }
        i++;
    }
}
```

```
        }

        // 将数据子集添加到图表数据集合中
        mDataset.addSeries(seriesSlave);

        // ColoursXYSeriesRenderer 是在 Achartengine 开源的项目中自定义的类
        ColoursXYSeriesRenderer rendererSlave = setSlaveSeriesRenderer(mChartPagerBean);

        // 将曲线渲染对象添加到图表渲染器中
        mRenderer.addSeriesRenderer(rendererSlave);

    }

    setMaxMinVallue(xAxisMax, mRenderer, yAxisMin, yAxisMax);
    layout.removeAllViews();
    mChartView = setView(context, mDataset, mRenderer);
    // 将图表视图添加图表容器中
    layout.addView(mChartView, new LayoutParams(LayoutParams.MATCH_PARENT,
                                                LayoutParams.MATCH_PARENT));
    // 图表重绘
    mChartView.repaint();
}

/* *
 * 设置图表 XY 轴的最大值和最小值
 * @param xAxisMax X 轴最大值
 * @param mRenderer 渲染器
 * @param yAxisMin Y 轴最小值
 * @param yAxisMax Y 轴最大值
 */
private void setMaxMinVallue(int xAxisMax,
                             XYMultipleSeriesRenderer mRenderer, int yAxisMin, int yAxisMax) {
    mRenderer.setYAxisMin(yAxisMin);
    mRenderer.setYAxisMax(yAxisMax);
    mRenderer.setXAxisMin(1);
    mRenderer.setXAxisMax(xAxisMax + 1);
}

/* *
 * 创建图表 View
 * @param context 上下文对象
 * @param mDataset 数据集
 * @param mRenderer 渲染器
 * @return 图表视图
 */
private GraphicalView setView(Context context,
                             XYMultipleSeriesDataset mDataset, XYMultipleSeriesRenderer mRenderer) {
```

```

GraphicalView mChartView;
mChartView = ChartFactory
    .getLineChartView(context, mDataset, mRenderer);
mRenderer.setApplyBackgroundColor(true); // 开启背景颜色
mRenderer.setBackgroundColor(Color.TRANSPARENT); // 设置背景透明
mRenderer.setMarginsColor(Color.TRANSPARENT); // 设置边缘透明
mRenderer.setSelectableBuffer(10);
return mChartView;
}
/*
* 创建次曲线渲染对象
* @param mChartPagerBean 图表容器
* @return 曲线渲染对象
*/
private ColoursXYSeriesRenderer setSlaveSeriesRenderer(
    ChartPagerBean mChartPagerBean) {
    ColoursXYSeriesRenderer rendererSlave = new ColoursXYSeriesRenderer();
    rendererSlave.setPointStyle(PointStyle.CIRCLE); // 设置为曲线图
    rendererSlave.setFillPoints(true); // 数据点被填充
    rendererSlave.setDisplayChartValues(true); // 在图表中显示点的值
    rendererSlave.setDisplayChartValuesDistance(1); // 在 X 轴上,设置两点之间的距离
    rendererSlave.setColor(Color.YELLOW); // 设置点的颜色
    rendererSlave.setChartValuesTextSize(25); // 设置点值文本的尺寸大小
    rendererSlave.setChartValuesSpacing(15f); // 实时数据文本到中心点的间距
    rendererSlave.setUseColor(true); // 设置当前为双色曲线图
    rendererSlave.setWarningMinValue(mChartPagerBean.slaveWarningMin); // 设置告警最小值
    rendererSlave.setWarningMaxValue(mChartPagerBean.slaveWarningMax); // 设置告警最大值
    return rendererSlave;
}
/*
* 创建主曲线渲染对象
* @param mChartPagerBean 图表容器
* @return 曲线渲染对象
*/
private ColoursXYSeriesRenderer setMajorSeriesRenderer(
    ChartPagerBean mChartPagerBean) {
    ColoursXYSeriesRenderer rendererMajor = new ColoursXYSeriesRenderer();
    rendererMajor.setPointStyle(PointStyle.CIRCLE); // 设置为曲线图
    rendererMajor.setFillPoints(true); // 数据点被填充
    rendererMajor.setDisplayChartValues(true); // 在图表中显示点的值
    rendererMajor.setDisplayChartValuesDistance(1); // 在 X 轴上,设置两点之间的距离
}

```

```
rendererMajor.setColor(Color.BLUE); // 设置点的颜色
rendererMajor.setChartValuesTextSize(25); // 设置点值文本的尺寸大小
rendererMajor.setChartValuesSpacing(15f); // 实时数据文本到中心点的间距
rendererMajor.setUseColor(true); // 设置当前为双色曲线图
rendererMajor.setPointColor(Color.GREEN); // 设置点的颜色
rendererMajor.setChartValueTextColor(Color.WHITE); // 设置数值文本的颜色
rendererMajor.setWarningMinValue(mChartPagerBean.majorWarningMin); // 设置告警最小值
rendererMajor.setWarningMaxValue(mChartPagerBean.majorWarningMax); // 设置告警最大值
return rendererMajor;
}
/*
 * 创建图表数据渲染器
 * @return 图表数据渲染器
 */
private XYMultipleSeriesRenderer setRenderer() {
    XYMultipleSeriesRenderer mRenderer;
    mRenderer = new XYMultipleSeriesRenderer();
    mRenderer.setApplyBackgroundColor(true); // 开启背景颜色
    mRenderer.setAxisTitleTextSize(25); // 设置坐标文字尺寸大小
    mRenderer.setAxesColor(Color.BLACK); // 设置坐标文字颜色
    mRenderer.setChartTitleTextSize(20); // 设置图表标题文字尺寸大小
    mRenderer.setLabelsTextSize(18); // 设置刻度显示文字的大小(XY 轴都会被设置)
    mRenderer.setLabelsColor(Color.BLACK); // 设置 XY 轴线的颜色
    mRenderer.setLegendTextSize(25); // 图例文字大小
    mRenderer.setMargins(new int[] { 0, 20, 0, 20 }); // 设置图表的外边框(上/左/下/右)
    mRenderer.setZoomButtonsVisible(true); // 是否显示放大缩小按钮
    mRenderer.setPointSize(10); // 设置点的大小(图上显示的点的大小和图例中点的大小都会被设置)
    mRenderer.setXLabelsColor(Color.WHITE); // 设置 X 轴标签文本的颜色
    mRenderer.setYLabelsColor(0, Color.WHITE); // 设置 Y 轴标签文本的颜色
    return mRenderer;
}
```

(2) res/layout/activity_main.xml 中的核心代码

```
<? xml version="1.0" encoding="utf-8"? >
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/chart_show"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/app_bg"
    android:orientation="horizontal"
```

```
    android:padding= "15dp">

</LinearLayout>
(3)res/values/strings.xml 中的核心代码
<? xml version= "1.0" encoding= "utf-8"? >
<resources>

    <string name= "app_name">ChartShow</string>
    <string name= "temp">温度</string>

</resources>
(4)AndroidManifest.xml 中的核心代码
<? xml version= "1.0" encoding= "utf-8"? >
<manifest xmlns:android= "http://schemas.android.com/apk/res/android"
    package= "lenovo.agriculture.chartshow"
    android:versionCode= "1"
    android:versionName= "1.0" >

    <uses-sdk
        android:minSdkVersion= "15"
        android:targetSdkVersion= "21"/>

    <application
        android:allowBackup= "true"
        android:icon= "@drawable/ic_launcher"
        android:label= "@string/app_name"
        android:theme= "@style/AppTheme">
        <activity
            android:name= "lenovo.agriculture.chartshow.activity.MainActivity"
            android:label= "@string/app_name"
            android:screenOrientation= "landscape">
            <intent-filter>
                <action android:name= "android.intent.action.MAIN"/>

                <category android:name= "android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
    </application>

</manifest>
```

4.11.4 任务运作(Operate)

开发工程师之间两两交叉代码评审，并根据评审纪要重构代码。

所有团队成员召开评审会，并填写表 4-11-2。

表 4-11-2

评审建议表

序号	建议描述	是否接纳	原因
1		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
2		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
3		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
4		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
5		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
6		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
7		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
8		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
9		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
10		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	

所有团队成员召开复盘会，并填写表 4-11-3。

表 4-11-3

复盘会总结表

回顾目标		评估结果	
当初的目的是什么（期望的结果）		Highlights (与原来目标比)	
要达成的目标 & 里程碑		Lowlights (与原来目标比)	
Goal	1	Result	2
Insight	3	Analysis	4
总结规律			
经验 & 规律（不要轻易下结论）			
行动计划			
新举措：			
叫停：			
继续：			
分析原因			
成功关键因素（主观/客观）			
失败根本原因（主观/客观）			

同学们，根据塔克曼阶梯理论，你们的团队目前处于哪个阶段呢？如果还没有进入成熟阶段，那就让我们一起努力吧，在接下来的任务中，更加注重相互配合和协作，及时

发现并改正团队配合过程中出现的问题,早日让我们的团队进入成熟阶段!

4.11.5 任务扩展(Extend)

课后练习:使用 AChartEngine 绘制饼图和柱状图。

子任务 12 查询传感器在某个时间段内的历史数据

- 任务目标:

- 查询某个传感器在某个时间段内的历史数据

- 课时分配:

- 4 课时

4.12.1 任务构思(Conceive)

在联想智能农业系统中,客户端 App 可以查询沙盘上的各个传感器在某个时间段内的历史数据,并以折线图的形式进行显示。

本任务对应示例程序的运行效果如下:

(1)启动示例程序,如图 4-12-1 所示。

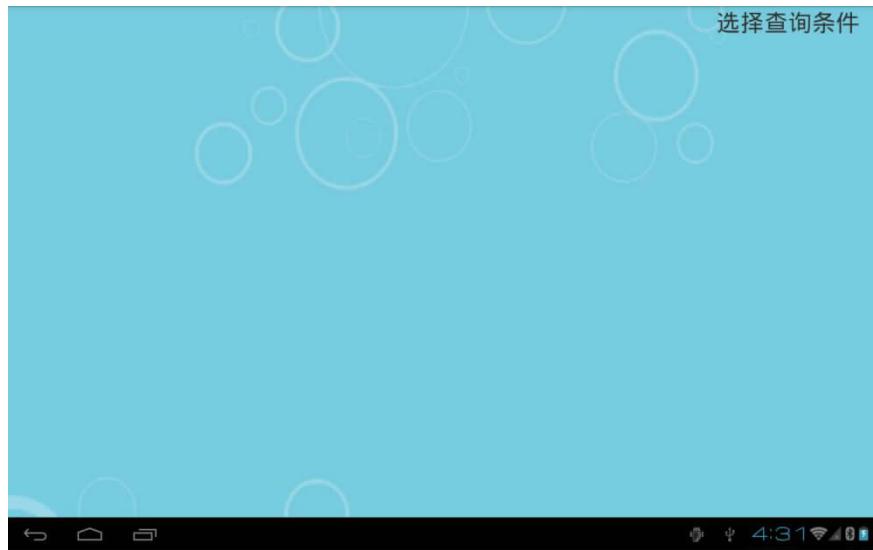


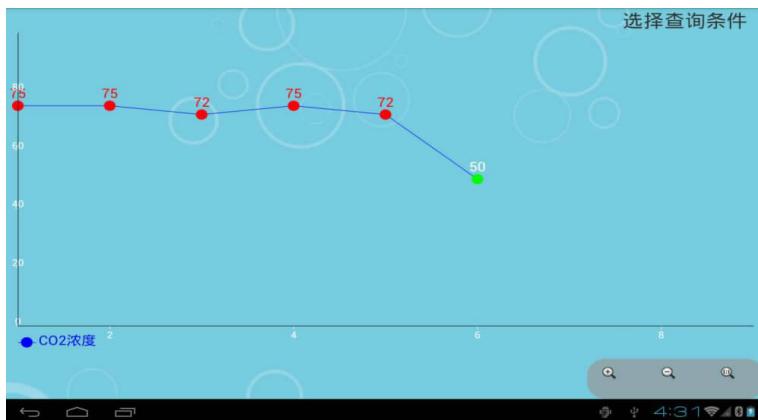
图 4-12-1 启动示例程序

(2)点击按钮【选择查询条件】,如图 4-12-2 所示。



图 4-12-2 设置查询条件

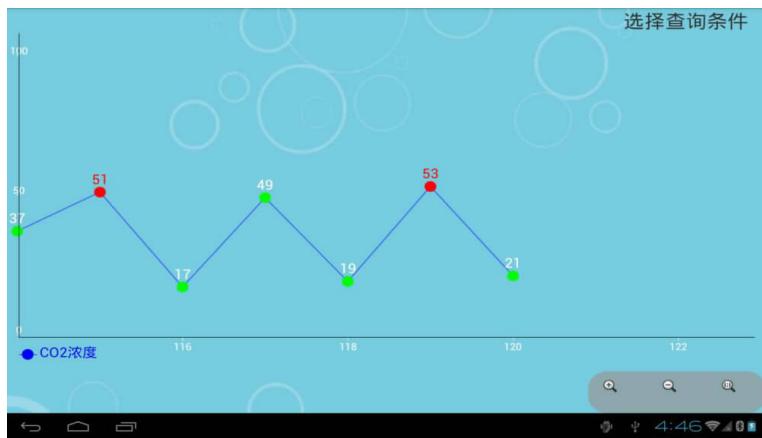
(3) 将查询的传感器选定为“CO₂”，查询时间选定为“一分钟内”，点击【OK】按钮，即可查询到 CO₂在一分钟内的浓度值，如图 4-12-3 所示。

图 4-12-3 查询 CO₂在一分钟内的浓度值

(4) 将查询的传感器选定为“CO₂”，查询时间选定为“五分钟内”，如图 4-12-4 所示，点击【OK】按钮，即可查询出 CO₂在五分钟内的浓度值，如图 4-12-5 所示。



图 4-12-4 设置查询条件

图 4-12-5 查询 CO₂在五分钟内的浓度值

(5) 将查询的传感器选定为“CO₂”，查询时间选定为“十分钟内”，如图 4-12-6 所示，点击【OK】按钮，即可查询出 CO₂在十分钟内的浓度值，如图 4-12-7 所示。



图 4-12-6 设置查询条件

图 4-12-7 查询 CO₂在十分钟内的浓度值

(6) 将查询的传感器选定为“光照”，查询时间选定为“一分钟内”，如图 4-12-8 所示，

点击【OK】按钮，即可查询出一分钟内的光照强度值，如图 4-12-9 所示。



图 4-12-8 设置查询条件

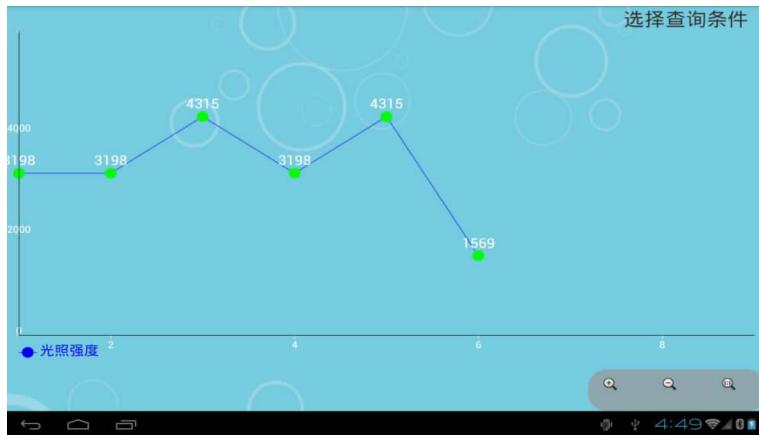


图 4-12-9 查询一分钟内的光照强度值

(7)将查询的传感器选定为“光照”，查询时间选定为“五分钟内”，如图 4-12-10 所示，点击【OK】按钮，即可查询出五分钟内的光照强度值，如图 4-12-11 所示。



图 4-12-10 设置查询条件

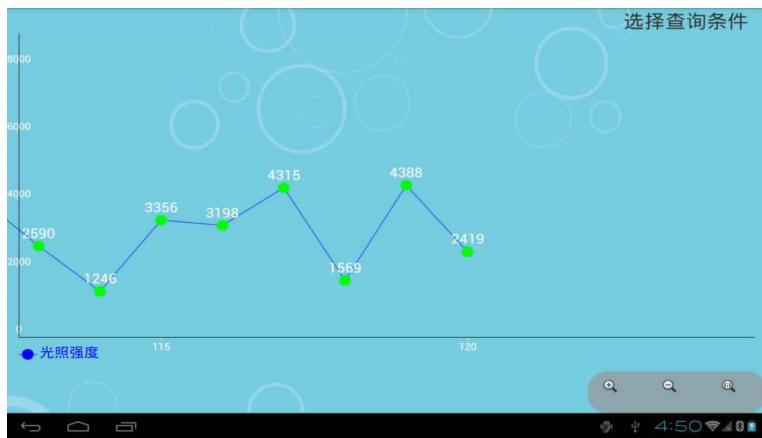


图 4-12-11 查询五分钟内的光照强度值

(8) 将查询的传感器选定为“光照”，查询时间选定为“十分钟内”，如图 4-12-12 所示，点击【OK】按钮，即可查询出十分钟内的光照强度值，如图 4-12-13 所示。



图 4-12-12 设置查询条件



图 4-12-13 查询十分钟内的光照强度值

(9) 将查询的传感器选定为“空气温湿度”，查询时间选定为“一分钟内”，如图 4-12-14 所示。

所示,点击【OK】按钮,即可查询出空气温湿度在一分钟内的值,如图 4-12-15 所示。



图 4-12-14 设置查询条件

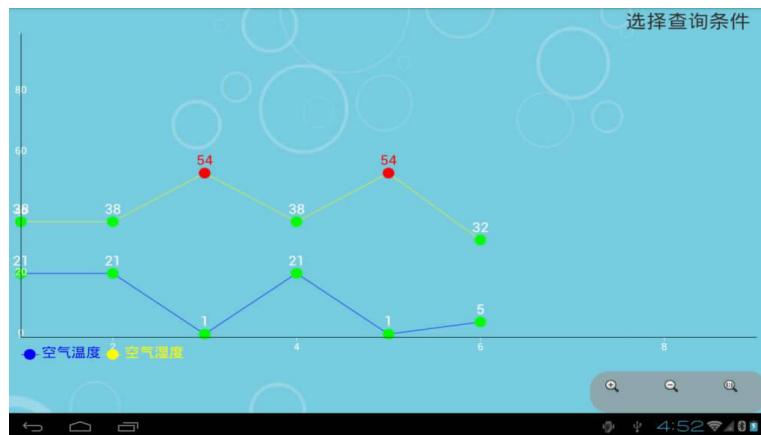


图 4-12-15 查询空气温湿度在一分钟内的值

(10)将查询的传感器选定为“空气温湿度”,查询时间选定为“五分钟内”,如图 4-12-16 所示,点击【OK】按钮,即可查询出空气温湿度在五分钟内的值,如图 4-12-17 所示。



图 4-12-16 设置查询条件

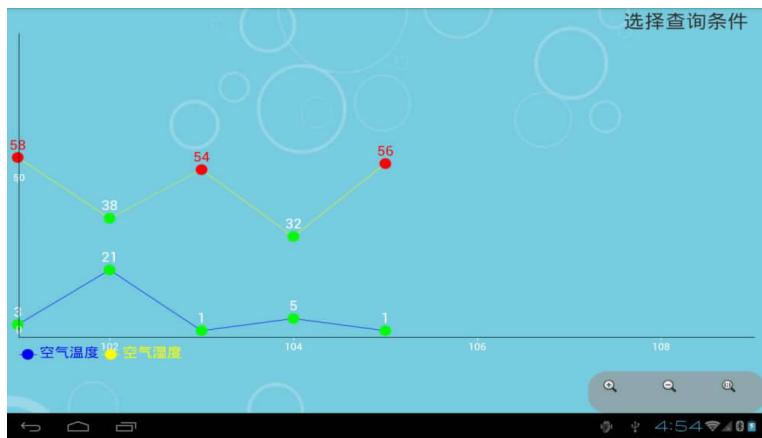


图 4-12-17 查询空气温湿度在五分钟内的值

(11) 将查询的传感器选定为“空气温湿度”，查询时间选定为“十分钟内”，如图 4-12-18 所示，点击【OK】按钮，即可查询出空气温湿度在十分钟内的值，如图 4-12-19 所示。



图 4-12-18 设置查询条件

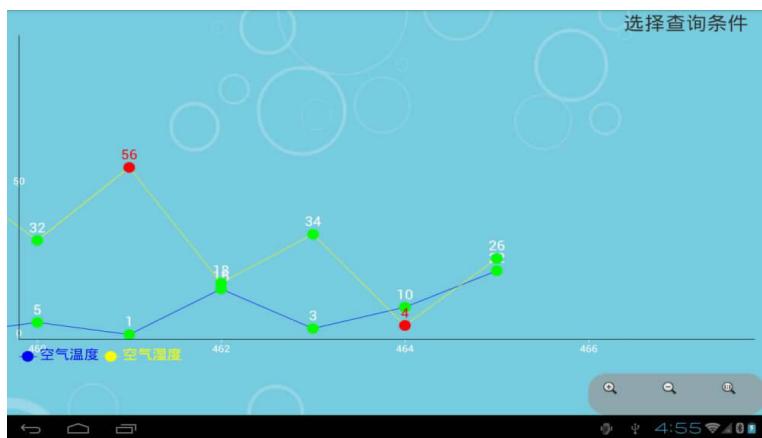


图 4-12-19 查询空气温湿度在十分钟内的值

(12) 将查询的传感器选定为“土壤温湿度”，查询时间选定为“一分钟内”，如图 4-12-

20 所示,点击【OK】按钮,即可查询出土壤温湿度在一分钟内的值,如图 4-12-21 所示。



图 4-12-20 设置查询条件

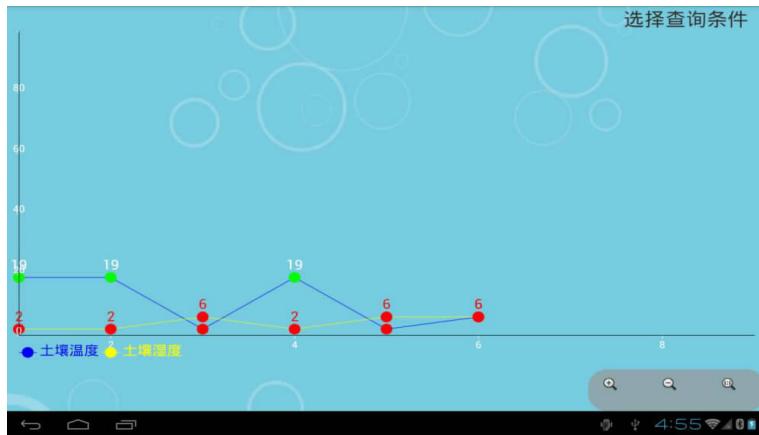


图 4-12-21 查询土壤温湿度在一分钟内的值

(13)将查询的传感器选定为“土壤温湿度”,查询时间选定为“五分钟内”,如图 4-12-22 所示,点击【OK】按钮,即可查询出土壤温湿度在五分钟内的值,如图 4-12-23 所示。



图 4-12-22 设置查询条件

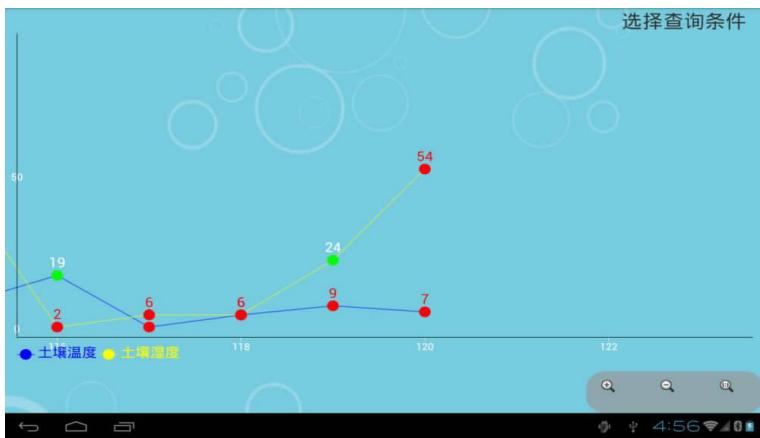


图 4-12-23 查询土壤温湿度在五分钟内的值

(14) 将查询的传感器选定为“土壤温湿度”，查询时间选定为“十分钟内”，如图 4-12-24 所示，点击【OK】按钮，即可查询出土壤温湿度在十分钟内的值，如图 4-12-25 所示。



图 4-12-24 设置查询条件

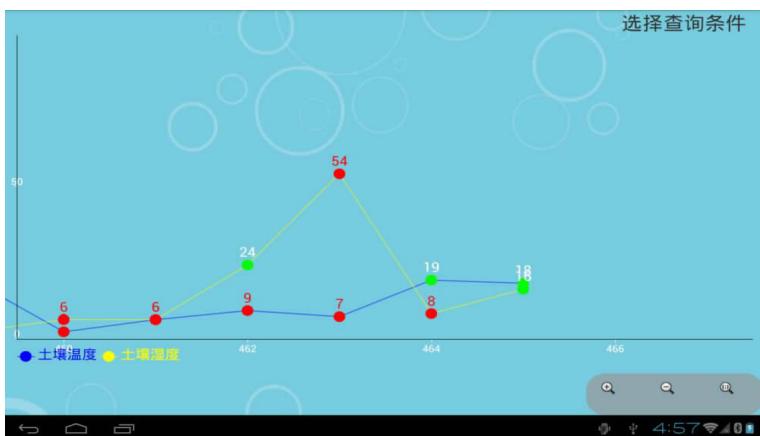


图 4-12-25 查询土壤温湿度在十分钟内的值

4.12.2 任务设计(Design)

本任务对应示例程序的主要设计思路为：

- (1)服务端将各个传感器的数据实时传递给客户端 App 并保存在 SQLite 数据库中；
- (2)客户端 App 根据查询条件从 SQLite 数据库中查询历史数据；
- (3)客户端 App 通过 AChartEngine 把查询到的历史数据以折线图的形式进行显示。

4.12.3 任务实现(Implement)

所有团队成员召开计划会议，并将会议纪要记录在表 4-12-1 中。

表 4-12-1

会议纪要记录表

5W2H	结论
WHAT	
HOW	
WHY	
WHEN	
WHERE	
WHO	
HOW MUCH	

在计划会议结束之前，所有团队成员把子任务进一步细分为更小的任务，把这些任务写在便签上，并把便签贴在看板上的“ToDo”列。

所有团队成员每天早上召开 15 分钟的站立会议，每人汇报三个问题，并更新看板上的“ToDo”、“Doing”和“Done”三列。

本任务对应示例程序的核心代码如下：

(1)lenovo.agriculture.querydata.db.BaseDB 类中的核心代码

```
/*
 * 获取数据
 * @param list 存储读出来的数据
 * @param selection 读取数据的限制条件
 * @return 是否查询成功
 */
private boolean loadDataFromDBEnter(ArrayList<Object> list, String selection) {
    boolean ret = true;
    // 调用读取之前的回调函数
    onLoadDataFromDBbefore();
    if (list != null) {
        Cursor contactCursor = null;
        try {
```

```

contactCursor = mDBHelper.getReadableDatabase().query(true ,
    getTableName(), null , selection, null , null , null ,
    null , null );
if (contactCursor != null ) {
    if (contactCursor.moveToFirst()) {
        do {
            // 从游标中解析出 java 对象并插入到 list 中
            Object data = getDataByCursor(contactCursor);
            if (data != null ) {
                list.add(data);
            }
        } while (contactCursor.moveToNext());
    }
}
} catch (Exception e) {
    ret = false ;
    e.printStackTrace();
} finally {
    if (contactCursor != null ) {
        contactCursor.close();
    }
}
}

// 调用读取之后的回调函数
onLoadDataFromDBAfter();
return ret;
}

```

(2) lenovo.agriculture.querydata.activity.MainActivity 类中的核心代码

```

/*
 * 显示可选择传感器和时间的对话框
 */
private void initDialog() {
    mSensorView.setSelention(sensorindex);
    mTimerView.setSelention(timeindex);
    mBuilder = new AlertDialog.Builder(MainActivity.this );
    mBuilder.setTitle(getString(R.string.choose_titl )).setView(mView)
        .setPositiveButton("OK", new DialogInterface.OnClickListener() {

    @Override
    public void onClick(DialogInterface dialog, int which) {
        mDb.loadSensorByTime(

```

```
new Date().getTime() - currentTime,  
new OnDatabaseHandlerListener() {  
  
    @Override  
    public boolean onDatabaseHandler(  
        ArrayList<Object> dataList,  
        boolean isSuccess) {  
        if (isSuccess) {  
            for (int i = 0; i < mBeans.size(); i++) {  
                mBeans.get(i).majorValueList  
                    .clear();  
                if (mBeans.get(i).isHasSlave) {  
                    mBeans.get(i).slaveValueList  
                        .clear();  
                }  
            }  
            for (int i = 0; i < dataList.size(); i++) {  
                SensorValue value = (SensorValue) dataList  
                    .get(i);  
                mBeans.get(0).majorValueList  
                    .add(value.getCo2());  
                mBeans.get(1).majorValueList  
                    .add(value.getLight());  
                mBeans.get(2).majorValueList.add(value  
                    .getAirTemper());  
                mBeans.get(2).slaveValueList.add(value  
                    .getAirHumid() * 2);  
                mBeans.get(3).majorValueList.add(value  
                    .getSoilTemper());  
                mBeans.get(3).slaveValueList.add(value  
                    .getSoilHumid() * 2);  
            }  
        }  
        mChartView.draw(MainActivity.this, mBean,  
            (LinearLayout) findViewById(R.id.chart_show), 8);  
        return false;  
    }  
});  
});  
}
```

(3) res/layout/activity_main.xml 中的核心代码

```
<? xml version= "1.0" encoding= "utf-8"? >
<LinearLayout xmlns:android= "http://schemas.android.com/apk/res/android"
    xmlns:tools= "http://schemas.android.com/tools"
    android:layout_width= "match_parent"
    android:layout_height= "match_parent"
    android:background= "@drawable/app_bg"
    android:orientation= "vertical">

    <TextView
        android:id= "@+id/show_wheel"
        android:layout_width= "match_parent"
        android:layout_height= "wrap_content"
        android:layout_marginRight= "30dp"
        android:gravity= "end"
        android:text= "@string/choose_text"
        android:textSize= "35sp"/>

    <LinearLayout
        android:id= "@+id/chart_show"
        android:layout_width= "match_parent"
        android:layout_height= "wrap_content"
        android:orientation= "vertical">
    </LinearLayout>

</LinearLayout>
```

(4) res/values/strings.xml 中的核心代码

```
<? xml version= "1.0" encoding= "utf-8"? >
<resources>

    <string name= "app_name">QueryData</string>
    <string name= "choose_text">选择查询条件</string>
    <string name= "sensor_co2">CO2</string>
    <string name= "sensor_light">光照</string>
    <string name= "sensor_air">空气温湿度</string>
    <string name= "sensor_soil">土壤温湿度</string>
    <string name= "time_1">一分钟内</string>
    <string name= "time_5">五分钟内</string>
    <string name= "time_10">十分钟内</string>
    <string name= "co2">CO2 浓度</string>
    <string name= "light">光照强度</string>
```

```
<string name= "air_t">空气温度</string>
<string name= "air_h">空气湿度</string>
<string name= "soil_t">土壤温度</string>
<string name= "soil_h">土壤湿度</string>
<string name= "choose_titl">请选择要查询的传感器和时间</string>

</resources>
(5)AndroidManifest. xml 中的核心代码
<? xml version= "1. 0" encoding= "utf-8"? >
<manifest xmlns:android= "http://schemas.android.com/apk/res/android"
    package= "lenovo.agriculture.querydata"
    android:versionCode= "1"
    android:versionName= "1. 0" >

    <uses-sdk
        android:minSdkVersion= "15"
        android:targetSdkVersion= "21"/>

    <application
        android:allowBackup= "true"
        android:icon= "@drawable/ic_launcher"
        android:label= "@string/app_name"
        android:theme= "@style/AppTheme">
        <activity
            android:name= ".activity.MainActivity"
            android:label= "@string/app_name"
            android:screenOrientation= "landscape">
            <intent-filter>
                <action android:name= "android.intent.action.MAIN"/>

                <category android:name= "android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
    </application>

</manifest>
```

4. 12. 4 任务运作(Operate)

开发工程师之间两两交叉代码评审，并根据评审纪要重构代码。

所有团队成员召开评审会，并填写表 4-12-2。

表 4-12-2

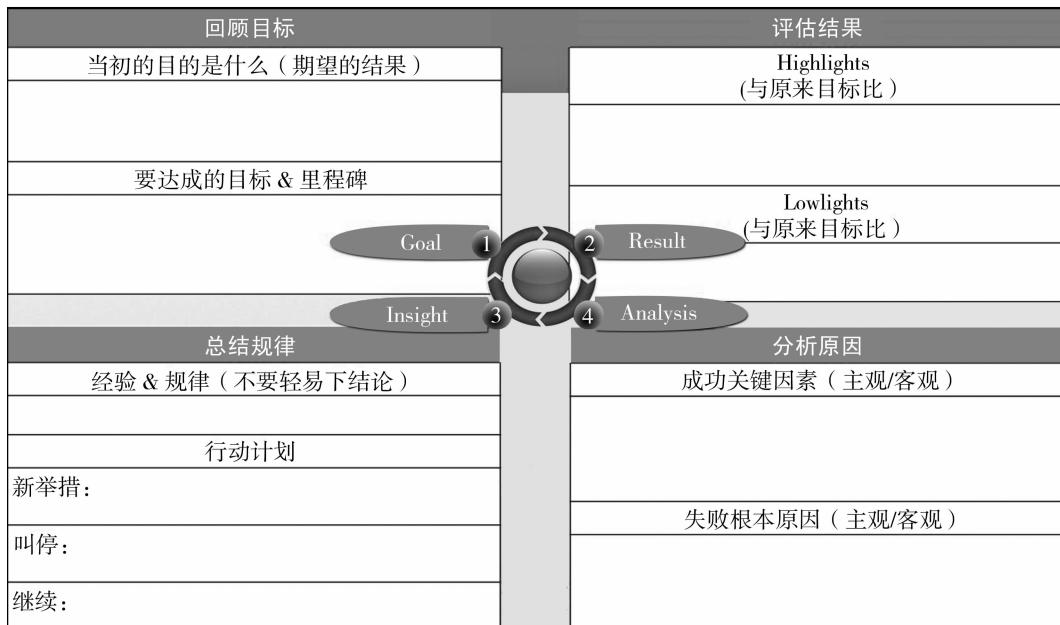
评审建议表

序号	建议描述	是否接纳	原因
1		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
2		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
3		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
4		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
5		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
6		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
7		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
8		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
9		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
10		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	

所有团队成员召开复盘会，并填写表 4-12-3。

表 4-12-3

复盘会总结表



同学们，根据塔克曼阶梯理论，你们的团队目前处于哪个阶段呢？如果还没有进入成熟阶段，那就让我们一起努力吧，在接下来的任务中，更加注重相互配合和协作，及时发现并改正团队配合过程中出现的问题，早日让我们的团队进入成熟阶段！

4.12.5 任务扩展(Extend)

课后练习：在该任务对应示例程序的基础上，把查询到的历史数据以柱状图的形式进行显示。

子任务 13 实现系统的权限管理

- 任务目标：

- 实现系统的权限管理

- 课时分配：

- 4 课时

4.13.1 任务构思(Conceive)

在联想智能农业系统中，管理员拥有系统的所有权限，而普通用户只拥有系统的部分权限。

本任务对应示例程序的运行效果如下：

(1)启动示例程序，进入程序的权限管理界面，如图 4-13-1 所示。



图 4-13-1 程序的权限管理界面

(2)点击按钮【打开】，如图 4-13-2 所示。



图 4-13-2 【打开】风扇

(3)添加新用户 admin1,如图 4-13-3 所示。



图 4-13-3 添加新用户 admin1

(4)选择 admin1 的角色,如图 4-13-4 所示。



图 4-13-4 选择 admin1 的角色

(5)给 admin1 赋予【普通用户】的角色后,如图 4-13-5 所示。



图 4-13-5 给 admin1 赋予【普通用户】角色

(6)选择刚添加的 admin1 用户,如图 4-13-6 所示。



图 4-13-6 选择刚添加的 admin1 用户

(7)点击按钮【打开】,如图 4-13-7 所示。



图 4-13-7 点击按钮【打开】

(8)点击【选择权限】下拉菜单,修改 admin1 的角色,如图 4-13-8 所示。



图 4-13-8 修改 admin1 的角色

(9)点击按钮【保存】，保存刚才修改的状态，点击界面右侧的按钮【打开】，如图 4-13-9 所示。



图 4-13-9 保存修改状态并【打开】风扇

4.13.2 任务设计(Design)

本任务对应示例程序的主要设计思路：

- (1) 创建多个角色，为不同的角色赋予不同的权限；
- (2) 创建多个用户，为不同的用户赋予不同的角色；
- (3) 根据用户的角色，判断其是否具有某个权限。

4.13.3 任务实现(Implement)

所有团队成员召开计划会议，并将会议纪要记录在表 4-13-1 中。

表 4-13-1 会议纪要记录表

5W2H	结论
WHAT	
HOW	
WHY	
WHEN	
WHERE	
WHO	
HOW MUCH	

在计划会议结束之前，所有团队成员把子任务进一步细分为更小的任务，把这些任务写在便签上，并把便签贴在看板上的“ToDo”列。

所有团队成员每天早上召开 15 分钟的站立会议，每人汇报三个问题，并更新看板上的“ToDo”、“Doing”和“Done”三列。

本任务对应示例程序的核心代码如下：

(1)lenovo.agriculture.authoritymanagement.MainActivity类中的核心代码

```
/*
 * 角色管理
 * @param role 角色
 */
private void authorityChoose(String role) {
    if (getString(R.string.manager).equals(role)) {
        if (clickNum % 2 == 0) {
            mPictureIV.setImageResource(R.drawable.open_icon);
            mOpenBtn.setBackgroundResource(R.drawable.close_btn);
        } else {
            mPictureIV.setImageResource(R.drawable.close_icon);
            mOpenBtn.setBackgroundResource(R.drawable.open_btn);
        }
        clickNum++;
    } elseif (getString(R.string.general_user).equals(role)) {
        showAlertDialog(getString(R.string.prompt), getString(R.string.msg));
    }
}
```

(2)res/layout/activity_main.xml中的核心代码

```
<? xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <LinearLayout
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="2"
        android:orientation="vertical">

        <EditText
            android:id="@+id/add_user"
            style="@style/FontSize"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="@string/hint">

        <requestFocus/>
    
```

```
</EditText>

<TextView
    android:id="@+id/textView2"
    style="@style/FontSize"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/check_role"/>

<Spinner
    android:id="@+id/choose_role"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>

<Button
    android:id="@+id/ok"
    style="@style/FontSize"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:onClick="saveData"
    android:text="@string/ok"/>

<TextView
    android:id="@+id/textView1"
    style="@style/FontSize"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/check_user"/>

<Spinner
    android:id="@+id/choose_user"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"/>
</LinearLayout>
<LinearLayout
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="4"
    android:background="@drawable/app_bg"
    android:orientation="vertical">
```

```
<ImageView  
    android:id= "@+id/show"  
    android:layout_width= "match_parent"  
    android:layout_height= "0dp"  
    android:layout_weight= "4"  
    android:src= "@drawable/close_icon"/>  
  
<Button  
    android:id= "@+id/open_btn"  
    android:layout_width= "wrap_content"  
    android:layout_height= "0dp"  
    android:layout_gravity= "center_horizontal"  
    android:layout_weight= "1"  
    android:background= "@drawable/open_btn"  
    android:onClick= "pictureShow"/>  
</LinearLayout>  
  
</LinearLayout>  
(3)res/layout/item.xml 中的核心代码  
<? xml version= "1.0" encoding= "utf-8"? >  
<TextView xmlns:android= "http://schemas.android.com/apk/res/android"  
    android:id= "@+id/user"  
    style= "@style/FontSize"  
    android:layout_width= "match_parent"  
    android:layout_height= "match_parent"  
    android:orientation= "vertical"  
    android:padding= "5dp">  
  
</TextView>  
(4)res/values/array.xml 中的核心代码  
<? xml version= "1.0" encoding= "utf-8"? >  
<resources>  
  
    <string-array name= "roles">  
        <item>管理员</item>  
        <item>普通用户</item>  
    </string-array>  
  
</resources>  
(5)res/values/strings.xml 中的核心代码  
<? xml version= "1.0" encoding= "utf-8"? >
```

```
<resources>

    <string name= "app_name">AuthorityManagement</string>
    <string name= "hint">添加用户</string>
    <string name= "ok">保存</string>
    <string name= "manager">管理员</string>
    <string name= "general_user">普通用户</string>
    <string name= "prompt">提示</string>
    <string name= "msg">对不起,您没有权限! </string>
    <string name= "check_user">查看用户</string>
    <string name= "check_role">选择权限</string>

</resources>
```

(6) AndroidManifest.xml 中的核心代码

```
<? xml version= "1.0" encoding= "utf-8"? >
<manifest xmlns:android= "http://schemas.android.com/apk/res/android"
    package= "lenovo.agriculture.authoritymanagement"
    android:versionCode= "1"
    android:versionName= "1.0" >

    <uses-sdk
        android:minSdkVersion= "15"
        android:targetSdkVersion= "21"/>

    <application
        android:allowBackup= "true"
        android:icon= "@drawable/ic_launcher"
        android:label= "@string/app_name"
        android:theme= "@style/AppTheme">
        <activity
            android:name= ".activity.MainActivity"
            android:label= "@string/app_name"
            android:screenOrientation= "landscape"
            android:windowSoftInputMode= "stateHidden/adjustUnspecified">
            <intent-filter>
                <action android:name= "android.intent.action.MAIN"/>

                <category android:name= "android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
    </application>

</manifest>
```

4.13.4 任务运作(Operate)

开发工程师之间两两交叉代码评审，并根据评审纪要重构代码。

所有团队成员召开评审会，并填写表 4-13-2。

表 4-13-2

评审建议表

序号	建议描述	是否接纳	原因
1		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
2		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
3		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
4		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
5		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
6		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
7		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
8		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
9		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
10		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	

所有团队成员召开复盘会，并填写表 4-13-3。

表 4-13-3

复盘会总结表

回顾目标		评估结果	
当初的目的是什么（期望的结果）		Highlights (与原来目标比)	
要达成的目标 & 里程碑		Lowlights (与原来目标比)	
Goal	1	2	Result
Insight	3	4	Analysis
分析原因			
总结规律		成功关键因素（主观/客观）	
经验 & 规律（不要轻易下结论）			
行动计划			
新举措：			
叫停：		失败根本原因（主观/客观）	
继续：			

同学们,根据塔克曼阶梯理论,你们的团队目前处于哪个阶段呢?如果还没有进入成熟阶段,那就让我们一起努力吧,在接下来的任务中,更加注重相互配合和协作,及时发现并改正团队配合过程中出现的问题,早日让我们的团队进入成熟阶段!

4.13.5 任务扩展(Extend)

课后练习:在该任务对应示例程序的基础上,添加更多的用户、角色和权限。

子任务 14 在服务端记录系统日志

- 任务目标:

- 在服务端记录系统日志

- 课时分配:

- 4 课时

4.14.1 任务构思(Conceive)

在联想智能农业系统中,可以通过日志文件的形式将系统的重要操作记录下来。

本任务对应示例程序的运行效果如下:

(1)启动示例程序,进入服务端示例程序的主界面,如图 4-14-1 所示。



图 4-14-1 程序主界面

(2)将服务端的状态置为打开,如图 4-14-2 所示。



图 4-14-2 将服务端的状态置为打开

(3)启动客户端 App,然后进行注册登录操作。点击服务器端的按钮【显示日志】,打开日志界面,如图 4-14-3、图 4-14-4 和图 4-14-5 所示。

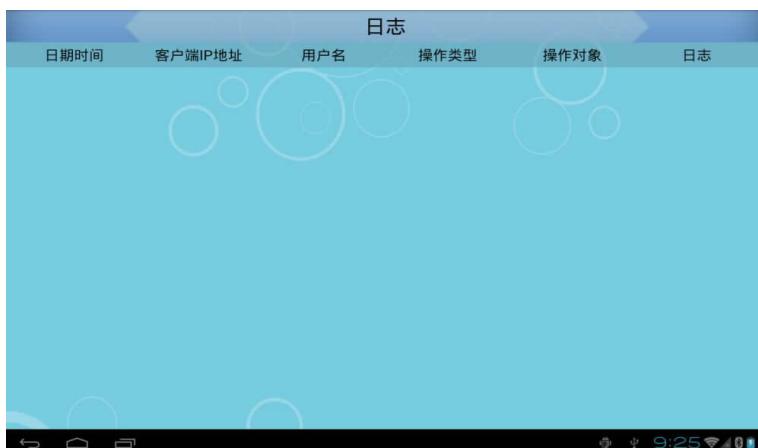


图 4-14-3 日志界面 1

日期时间	客户端IP地址	用户名	操作类型	操作对象	日志
2015-07-22 09:40:15	192.168.1.128	admin	jason	admin	User login success
2015-07-22 09:40:15	192.168.1.128	admin	jason	All Sensor	Get sensor value
2015-07-22 09:40:15	192.168.1.128	admin	jason	All Sensor	Get threshold value of sensor
2015-07-22 09:40:16	192.168.1.128	admin	jason	All Sensor	Get sensor value
2015-07-22 09:40:17	192.168.1.128	admin	jason	All Sensor	Get sensor value
2015-07-22 09:40:18	192.168.1.128	admin	jason	All Sensor	Get sensor value
2015-07-22 09:40:19	192.168.1.128	admin	jason	All Sensor	Get sensor value

图 4-14-4 日志界面 2

日志						
日期时间	客户端IP地址	用户名	操作类型	操作对象	日志	
2015-07-22 09:42:42	192.168.1.128	admin	jason	All Controller	Get controller status	
2015-07-22 09:42:50	192.168.1.128	admin	jason	All Controller	Get controller status	
2015-07-22 09:42:52	192.168.1.128	admin	jason	Roadlamp	Open Roadlamp	
2015-07-22 09:42:53	192.168.1.128	admin	jason	All Controller	Get controller status	
2015-07-22 09:42:54	192.168.1.128	admin	jason	Roadlamp	Close Roadlamp	
2015-07-22 09:42:55	192.168.1.128	admin	jason	All Controller	Get controller status	
2015-07-22 09:42:59	192.168.1.128	admin	jason	All Sensor	Get sensor value	
2015-07-22 09:43:00	192.168.1.128	admin	jason	All Sensor	Get sensor value	
2015-07-22 09:43:01	192.168.1.128	admin	jason	All Sensor	Get sensor value	

图 4-14-5 日志界面 3

4.14.2 任务设计(Design)

本任务对应示例程序的主要设计思路为：

- (1) 通过输入流将重要操作对应的日志信息写到日志文件中；
- (2) 通过输出流从日志文件中读出日志信息并以表格的形式进行显示。

4.14.3 任务实现(Implement)

所有团队成员召开计划会议，并将会议纪要记录在表 4-14-1 中。

表 4-14-1

会议纪要记录表

5W2H	结论
WHAT	
HOW	
WHY	
WHEN	
WHERE	
WHO	
HOW MUCH	

在计划会议结束之前，所有团队成员把子任务进一步细分为更小的任务，把这些任务写在便签上，并把便签贴在看板上的“ToDo”列。

所有团队成员每天早上召开 15 分钟的站立会议，每人汇报三个问题，并更新看板上的“ToDo”、“Doing”和“Done”三列。

本任务对应示例程序的核心代码如下：

(1) com.lenovo.logprint.service.NetServerService 类中的核心代码

```
/ * *
```

```
* 接收广播存储日志
*/
class LogServiceBroadcast extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        // 用来接收日志信息
        String message = intent.getStringExtra("log_message");
        if (message == null
            || (message.equals(mPreLogStr) && mPreLogStr
                .indexOf("[getSensor]") >= 0)) {
            return ;
        }
        mPreLogStr = message;
        FileOutputStream outStream;
        try {
            outStream = new FileOutputStream(mLogFile, true );
            outStream.write(message.getBytes());
            outStream.close();
        } catch (FileNotFoundException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

(2)com.lenovo.logprint.ui.LogListActivity类中的核心代码

```
/*
 * 另开线程从文本文件中读取日志内容
 */
private void loadLogDataFromFile() {
    new Thread() {
        @Override
        public void run() {
            super.run();
            FileInputStream fileIs = null ;
            ArrayList<LogBean> dataLists = new ArrayList<LogBean>();
            try {
                // 创建文件输入流
                fileIs = new FileInputStream(AppConfig.LOG_FILE_PATH );
                // 创建 reader buffer
                BufferedReader buf = new BufferedReader(
```

第4章 项目实现（Implement）——编码和测试

```
    new InputStreamReader(fileIs));
    String readString = "";
    // 一行一行地读取日志文本
    while((readString = buf.readLine()) != null) {
        // 以"]["为分隔符,取出日志各字段内容
        String logInfos[] = readString.split("\\]\\\\");

        LogBean logBean = new LogBean();
        logBean.setDate(logInfos[0].replace("[", ""));
        logBean.setClientIP(logInfos[1]);
        logBean.setUsername(logInfos[2]);
        logBean.setOperationType(logInfos[3]);
        logBean.setOperationObj(logInfos[4]);
        logBean.setLogMsg(logInfos[5].replace("]", ""));
        dataLists.add(logBean);
    }
    buf.close();
} catch (FileNotFoundException e) {
    e.printStackTrace();
} catch (PatternSyntaxException e) {
    e.printStackTrace();
} catch (IOException e) {
    e.printStackTrace();
} finally {
    try {
        if (fileIs != null) {
            // 关闭文件输入流
            fileIs.close();
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}
final ArrayList<LogBean> lists = dataLists;
runOnUiThread(new Runnable() {
    @Override
    public void run() {
        mDataLists.clear();
        mDataLists.addAll(lists);
        // 刷新列表
        mDataAdapter.notifyDataSetChanged();
        // 关闭对话框
    }
})
```

```
        dismissLoadDialog();
    }
}
);
}
}.start();
}

(3)res/layout/log_list.xml 中的核心代码
<? xml version="1.0" encoding="utf-8"? >
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/app_bg"
    android:orientation="vertical">

    <include layout="@layout/title_layout"/>

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:paddingBottom="5dp">

        <include layout="@layout/list_head"/>

        <ListView
            android:id="@+id/list"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:cacheColorHint="@color/transparent_background"
            android:fadingEdge="none"
            android:listSelector="@color/transparent_background"
            android:scrollingCache="false"/>
    </LinearLayout>

</LinearLayout>

(4)res/layout/list_head.xml 中的核心代码
<? xml version="1.0" encoding="utf-8"? >
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/list_head"
    android:layout_width="match_parent"
    android:layout_height="45dp"
```

```
    android:background= "@color/list_title_bg"
    android:gravity= "start/center_vertical"
    android:orientation= "horizontal"
    android:paddingLeft= "10dp"
    android:paddingRight= "10dp">

<TextView
    style= "@style/FontSize"
    android:layout_width= "0dp"
    android:layout_height= "wrap_content"
    android:layout_weight= "1"
    android:gravity= "center"
    android:text= "@string/date_time"
    android:textColor= "@color/black"/>

<TextView
    style= "@style/FontSize"
    android:layout_width= "0dp"
    android:layout_height= "wrap_content"
    android:layout_weight= "1"
    android:gravity= "center"
    android:text= "@string/client_ip"
    android:textColor= "@color/black"/>

<TextView
    style= "@style/FontSize"
    android:layout_width= "0dp"
    android:layout_height= "wrap_content"
    android:layout_weight= "1"
    android:gravity= "center"
    android:text= "@string/username"
    android:textColor= "@color/black"/>

<TextView
    style= "@style/FontSize"
    android:layout_width= "0dp"
    android:layout_height= "wrap_content"
    android:layout_weight= "1"
    android:gravity= "center"
    android:text= "@string/operation_type"
    android:textColor= "@color/black"/>
```

```
<TextView
    style="@style/FontSize"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:gravity="center"
    android:text="@string/operation_obj"
    android:textColor="@color/black"/>

<TextView
    style="@style/FontSize"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:gravity="center"
    android:text="@string/log"
    android:textColor="@color/black"/>

</LinearLayout>
(5)res/layout/list_item.xml 中的核心代码
<? xml version="1.0" encoding="utf-8"? >
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@android:color/transparent"
    android:cacheColorHint="@android:color/transparent"
    android:orientation="horizontal"
    android:paddingBottom="5dp"
    android:paddingLeft="10dp"
    android:paddingRight="10dp"
    android:paddingTop="5dp">

<TextView
    android:id="@+id/data_time_text_view"
    style="@style/FontSize"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:gravity="center"
    android:textColor="@color/black"/>
```

```
<TextView
    android:id="@+id/ip_text_view"
    style="@style/FontSize"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:gravity="center"
    android:textColor="@color/black"/>

<TextView
    android:id="@+id/username_text_view"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:gravity="center"
    android:textColor="@color/black"
    android:textSize="20sp"/>

<TextView
    android:id="@+id/operation_type_text_view"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:gravity="center"
    android:textColor="@color/black"
    android:textSize="20sp"/>

<TextView
    android:id="@+id/operation_obj_text_view"
    style="@style/FontSize"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:gravity="center"
    android:textColor="@color/black"/>

<TextView
    android:id="@+id/log_text_view"
    style="@style/FontSize"
    android:layout_width="0dp"
```

```
    android:layout_height= "match_parent"
    android:layout_weight= "1"
    android:gravity= "center"
    android:textColor= "@color/black"/>

</LinearLayout>
(6)res/layout/title_layout.xml 中的核心代码
<? xml version= "1.0" encoding= "utf-8"? >
<LinearLayout xmlns:android= "http://schemas.android.com/apk/res/android"
    android:layout_width= "match_parent"
    android:layout_height= "wrap_content">

<TextView
    android:id= "@+id/title_text_view"
    android:layout_width= "match_parent"
    android:layout_height= "60dp"
    android:layout_marginTop= "-3dp"
    android:background= "@drawable/title_bg"
    android:gravity= "center"
    android:text= "Title"
    android:textColor= "@color/black"
    android:textSize= "35sp"
    android:textStyle= "bold"/>

</LinearLayout>
(7)res/values/strings.xml 中的核心代码
<? xml version= "1.0" encoding= "utf-8"? >
<resources>

    <string name= "app_name">日志打印</string>
    <string name= "find_password_exp">联想农业服务系统,用户密码找回</string>
    <string name= "please_wait">请稍等</string>
    <string name= "date_time">日期时间</string>
    <string name= "client_ip">客户端 IP 地址</string>
    <string name= "username">用户名</string>
    <string name= "operation_type">操作类型</string>
    <string name= "operation_obj">操作对象</string>
    <string name= "log">日志</string>
    <string name= "status">服务状态:</string>
    <string name= "print_log">日志打印</string>
```

```
</resources>

(8) AndroidManifest.xml 中的核心代码
<? xml version= "1. 0" encoding= "utf-8"? >
<manifest xmlns:android= "http://schemas.android.com/apk/res/android"
    package= "com.lenovo.logprint"
    android:versionCode= "1"
    android:versionName= "1. 0" >

    <uses-sdk
        android:minSdkVersion= "14"
        android:targetSdkVersion= "19"/>

    <uses-permission android:name= "android.permission.WAKE_LOCK"/>
    <uses-permission android:name= "android.permission.INTERNET"/>
    <! -- 获取 Wifi 状态的权限 -->
    <uses-permission android:name= "android.permission.ACCESS_WIFI_STATE"/>
    <uses-permission android:name= "android.permission.CHANGE_WIFI_MULTICAST_STATE"/>
    <uses-permission android:name= "android.permission.ACCESS_NETWORK_STATE"/>
    <uses-permission android:name= "android.permission.WAKE_LOCK"/>
    <uses-permission android:name= "android.permission.WRITE_EXTERNAL_STORAGE"/>

    <application
        android:name= "com.lenovo.logprint.ServerApp"
        android:allowBackup= "true"
        android:icon= "@drawable/ic_launcher"
        android:label= "@string/app_name"
        android:persistent= "true"
        android:taskAffinity= "com.lenovo.logprint"
        android:theme= "@style/AppBaseTheme">
        <activity
            android:name= "com.lenovo.logprint.ui.MainActivity"
            android:label= "@string/app_name"
            android:screenOrientation= "landscape"
            android:windowSoftInputMode= "stateHidden/adjustUnspecified">
            <intent-filter>
                <action android:name= "android.intent.action.MAIN"/>

                <category android:name= "android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
        <activity
```

```

    android:name= "com.lenovo.logprint.ui.LogListActivity"
    android:label= "@string/app_name"
    android:screenOrientation= "landscape"
    android:windowSoftInputMode= "stateHidden/adjustUnspecified">
</activity>

<service
    android:name= "com.lenovo.logprint.service.NetServerService"
    android:enabled= "true"
    android:exported= "true">
    <intent-filter>
        <action android:name= "dlna.player.StartDMS"/>
        <action android:name= "dlna.player.StopDMS"/>
    </intent-filter>
</service>
<service
    android:name= "com.lenovo.logprint.Serial.SerialportService"
    android:enabled= "true"
    android:exported= "true"/>
</application>

</manifest>

```

4.14.4 任务运作(Operate)

开发工程师之间两两交叉代码评审，并根据评审纪要重构代码。

所有团队成员召开评审会，并填写表 4-14-2。

表 4-14-2

评审建议表

序号	建议描述	是否接纳	原因
1		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
2		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
3		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
4		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
5		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
6		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
7		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
8		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
9		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
10		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	

所有团队成员召开复盘会，并填写表 4-14-3。

表 4-14-3 复盘会总结表

回顾目标		评估结果	
当初的目的是什么（期望的结果）		Highlights (与原来目标比)	
要达成的目标 & 里程碑		Lowlights (与原来目标比)	
Goal	1	Result	2
Insight	3	Analysis	4
总结规律		分析原因	
经验 & 规律（不要轻易下结论）		成功关键因素（主观/客观）	
行动计划		失败根本原因（主观/客观）	
新举措：			
叫停：			
继续：			

同学们，根据塔克曼阶梯理论，你们的团队目前处于哪个阶段呢？如果还没有进入成熟阶段，那就让我们一起努力吧，在接下来的任务中，更加注重相互配合和协作，及时发现并改正团队配合过程中出现的问题，早日让我们的团队进入成熟阶段！

4.14.5 任务扩展(Extend)

课后练习：在该任务对应示例程序的基础上，通过工具 Log4J 记录系统日志。

子任务 15 使用 Monkey 对 App 进行压力测试

- 任务目标：

 - 使用 Monkey 对 App 进行压力测试

- 课时分配：

 - 4 课时

4.15.1 任务构思(Conceive)

在联想智能农业系统中，可以借助 Android 系统提供的压力测试工具对 App 进行压力测试。

本任务对应示例程序的运行效果如下：

(1) 将子任务 9 中的 App 安装成功后, 打开命令行, 检查设备连接, 如图 4-15-1 所示。



```
Microsoft Windows [版本 6.1.7601]
版权所有 © 2009 Microsoft Corporation。保留所有权利。
C:\Users\Schutg>adb devices
List of devices attached
0123456789ABCDEF      device

C:\Users\Schutg>
```

图 4-15-1 打开命令行

(2) 在命令行输入命令: adb shell monkey-p lenovo.agriculture.changelanguage-v 1000, 整个示例程序就像猴子一样不停地“跳来跳去”, 如图 4-15-2、图 4-15-3 和图 4-15-4 所示。

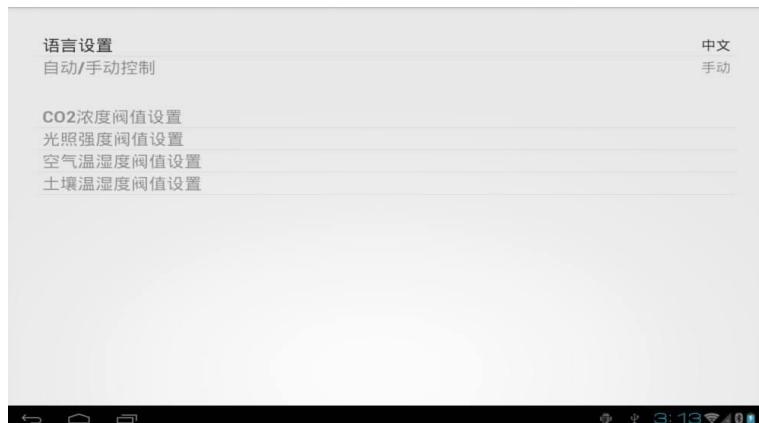


图 4-15-2 程序运行 1



图 4-15-3 程序运行 2

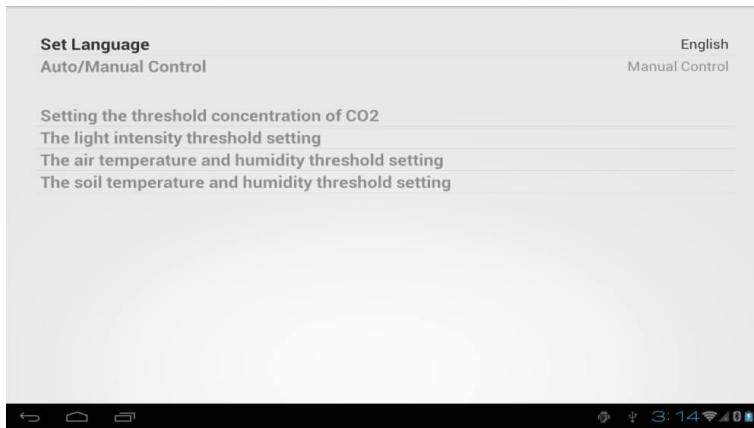


图 4-15-4 程序运行 3

4.15.2 任务设计(Design)

该任务对应示例程序的设计思路为：借助 Android 系统提供的压力测试工具 Monkey，随机地向应用程序发送事件流并指定事件流的个数，让应用程序处于某种压力测试状态。

4.15.3 任务实现(Implement)

所有团队成员召开计划会议，并将会议纪要记录在表 4-15-1 中。

表 4-15-1

会议纪要记录表

5W2H	结论
WHAT	
HOW	
WHY	
WHEN	
WHERE	
WHO	
HOW MUCH	

在计划会议结束之前，所有团队成员把子任务进一步细分为更小的任务，把这些任务写在便签上，并把便签贴在看板上的“ToDo”列。

所有团队成员每天早上召开 15 分钟的站立会议，每人汇报三个问题，并更新看板上的“ToDo”、“Doing”和“Done”三列。

4.15.4 任务运作(Operate)

所有团队成员召开评审会，并填写表 4-15-2。

表 4-15-2

评审建议表

序号	建议描述	是否接纳	原因
1		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
2		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
3		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
4		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
5		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
6		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
7		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
8		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
9		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
10		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	

所有团队成员召开复盘会，并填写表 4-15-3。

表 4-15-3

复盘会总结表

回顾目标		评估结果	
当初的目的是什么（期望的结果）		Highlights (与原来目标比)	
要达成的目标 & 里程碑		Lowlights (与原来目标比)	
Goal	1	2	Result
Insight	3	4	Analysis
分析原因			
总结规律		成功关键因素（主观/客观）	
经验 & 规律（不要轻易下结论）			
行动计划			
新举措：		失败根本原因（主观/客观）	
叫停：			
继续：			

同学们，根据塔克曼阶梯理论，你们的团队目前处于哪个阶段呢？如果还没有进入成熟阶段，那就让我们一起努力吧，在接下来的任务中，更加注重相互配合和协作，及时发现并改正团队配合过程中出现的问题，早日让我们的团队进入成熟阶段！

4.15.5 任务扩展(Extend)

课后练习：在该任务对应示例程序的基础上，使用 MonkeyRunner 替代 Monkey 进行压力测试。

第5章 项目运作(Operate)——签名和发布

- 本章目标：

- 应用 App 在发布之前的签名过程
- 应用 App 的发布过程

- 课时分配：

- 4 课时

5.1.1 任务构思(Conceive)

在智能农业实战项目中,如果要将应用程序发布到应用市场,需要对应用程序进行签名,以便和应用市场上的其他应用程序进行区分。

5.1.2 任务设计(Design)

在 Android 系统中,如果两个应用程序的包名相同,系统会认为它们是同一个应用程序,然后再比较它们的签名,如果签名不一样,系统就会认为它们是两个不同的应用程序。因此首先要给应用程序进行签名。在 Android 系统中,有两种签名方式:命令行方式和 Eclipse 插件的方式,两者的原理是一样的,为了简单起见,我们可以采用 Eclipse 插件的方式。

5.1.3 任务实现(Implement)

导出需要进行签名的 Android 应用程序,如图 5-1-1 所示:

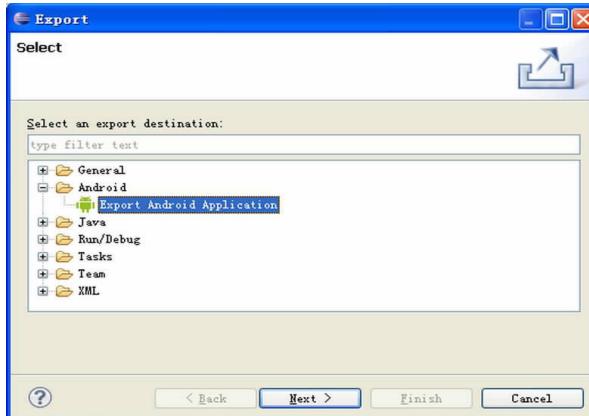


图 5-1-1 需要进行签名的 Android 应用程序

接下来按照提示,即可完成应用程序的签名。

应用程序签名后,接下来需要将其发布到 Google Play 或国内第三方应用市场,比如:联想乐商店、豌豆荚、360 等。应用市场的发布过程各不相同,可以到其官方网站了解具体的发布过程。

5.1.4 任务运作(Operate)

所有团队成员召开计划会议,并将会议纪要记录在表 5-1-1 中。

表 5-1-1

会议纪要记录表

5W2H	结论
WHAT	
HOW	
WHY	
WHEN	
WHERE	
WHO	
HOW MUCH	

在计划会议结束之前,所有团队成员把子任务进一步细分为更小的任务,把这些任务写在便签上,并把便签贴在看板上的“ToDo”列。

所有团队成员每天早上召开 15 分钟的站立会议,每人汇报三个问题,并更新看板上的“ToDo”、“Doing”和“Done”三列。

所有团队成员召开评审会,并填写表 5-1-2。

表 5-1-2

评审建议表

序号	建议描述	是否接纳	原因
1		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
2		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
3		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
4		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
5		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
6		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
7		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
8		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
9		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
10		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	

所有团队成员召开复盘会,并填写表 5-1-3。

表 5-1-3

复盘会总结表

回顾目标		评估结果	
当初的目的是什么 (期望的结果)		Highlights (与原来目标比)	
要达成的目标 & 里程碑		Lowlights (与原来目标比)	
Goal	1	Result	2
总结规律	Insight		3
经验 & 规律 (不要轻易下结论)		分析原因	
行动计划		成功关键因素 (主观/客观)	
新举措:		失败根本原因 (主观/客观)	
叫停:			
继续:			

同学们,根据塔克曼阶梯理论,你们的团队目前处于哪个阶段呢?如果还没有进入成熟阶段,那就让我们一起努力吧,在接下来的任务中,更加注重相互配合和协作,及时发现并改正团队配合过程中出现的问题,早日让我们的团队进入成熟阶段!

5.1.5 任务扩展(Extend)

课后练习:将应用程序重新签名并发布到联想乐商店应用市场。

第 6 章 项目扩展(Extend)——软件维护

- 本章目标：

- 应用“友盟”完成应用程序上线后的统计分析
- 应用 App Annie 完成应用程序上线后的统计分析

- 课时分配：

- 4 课时

6.1.1 任务构思(Conceive)

在智能农业实战项目中,当应用程序发布到应用市场后,需要对应用程序进行统计分析,以便及时调整和优化产品功能和营销方式等。应用程序可能需要具备分享、评论等社交功能,以便提升用户粘度、助力产品推广,因此需要实时、全面的社会化数据统计分析。此外,开发者需要建立与用户直接沟通的渠道,将 App 的内容更新或者活动通知主动推送给终端用户,让用户第一时间获取到相关信息,有效提升用户活跃度和忠诚度。

6.1.2 任务设计(Design)

为了对应用程序进行统计分析,需要借助第三方的服务平台,有如下两个比较知名的平台：

- (1) 友盟；
- (2) App Annie。

因此,我们可以分别利用这两个平台对上线后的智能农业实战项目进行统计分析。

6.1.3 任务实现(Implement)

友盟的统计分析功能如下：

- (1) 应用趋势分析如图 6-1-1 所示。

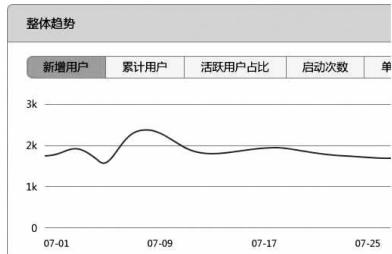


图 6-1-1 应用趋势分析

(2) 渠道分析如图 6-1-2 所示。

渠道质量度			
日期	累计用户	老用户占比	平均日
AndroidMarket	44938	18.7%	6753
Anzhi	37555	19.6%	7936
GooglePlay	26182	10.9%	4232
Nduo	25503	16.3%	2861

图 6-1-2 渠道分析

(3) 留存分析如图 6-1-3 所示。

留存用户			日 周 月		
首次使用时间	新增用户	留存率	1天后	2天后	3天后
2014-02-07	2745	40.6%	30.8%	25.8%	
2014-02-08	2696	40.5%	30.5%	26.6%	
2014-02-09	2711	40.6%	29.9%	25.9%	

图 6-1-3 留存分析

(4) 行为分析如图 6-1-4 所示。

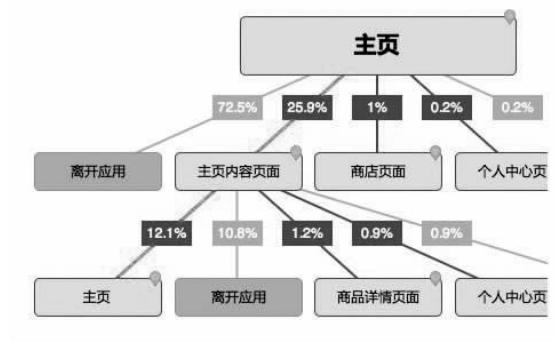


图 6-1-4 行为分析

(5) 用户属性分析如图 6-1-5 所示。

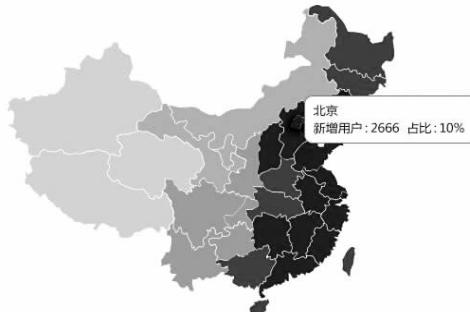


图 6-1-5 用户属性分析

(6) 错误分析如图 6-1-6 所示。



图 6-1-6 错误分析

App Annie 的统计分析功能如下：

(1) 下载量分析如图 6-1-7 所示。

Country	Downloads
Total	115,279 ↓ <0.5%
United States	35,106 ↑ <0.5%
China	11,564 ↓ <0.5%
Great Britain (UK)	11,099 ↓ 6%

图 6-1-7 下载量分析

(2) 收益分析如图 6-1-8 所示。



图 6-1-8 收益分析

(3) 排名分析如图 6-1-9 所示。

	Overall	Games
rank 1 reached	3 ▼2	5 ▼5
rank 5 reached	28 ▲4	64 ▲16
rank 10 reached	59 ▲15	97 ▲14
rank 100 reached	127 ▲2	149 =

图 6-1-9 排名分析

(4) 评论分析如图 6-1-10 所示。



图 6-1-10 评论分析

(5) 推荐分析如图 6-1-11 所示。

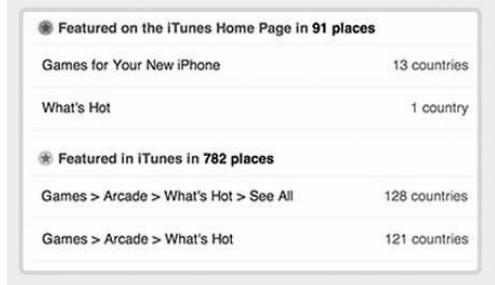


图 6-1-11 推荐分析

(6) 项目追踪分析如图 6-1-12 所示。

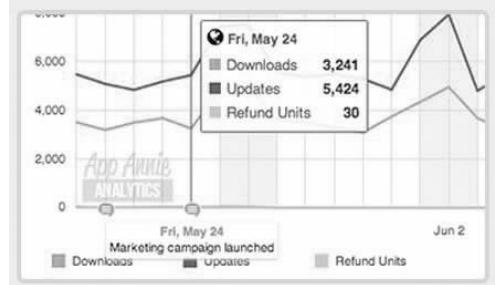


图 6-1-12 项目追踪分析

6.1.4 任务运作(Operate)

所有团队成员召开计划会议，并将会议纪要记录在表 6-1-1 中。

表 6-1-1

会议纪要记录表

5W2H	结论
WHAT	
HOW	
WHY	
WHEN	
WHERE	
WHO	
HOW MUCH	

在计划会议结束之前，所有团队成员把子任务进一步细分为更小的任务，把这些任务写在便签上，并把便签贴在看板上的“ToDo”列。

所有团队成员每天早上召开 15 分钟的站立会议，每人汇报三个问题，并更新看板上的“ToDo”、“Doing”和“Done”三列。

所有团队成员召开评审会，并填写表 6-1-2。

表 6-1-2

评审建议表

序号	建议描述	是否接纳	原因
1		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
2		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
3		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
4		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
5		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
6		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
7		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
8		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
9		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	
10		<input type="checkbox"/> 接纳 <input type="checkbox"/> 部分接纳 <input type="checkbox"/> 未接纳	

所有团队成员召开复盘会，并填写表 6-1-3。

表 6-1-3

复盘会总结表

回顾目标		评估结果	
当初的目的是什么 (期望的结果)		Highlights (与原来目标比)	
要达成的目标 & 里程碑		Lowlights (与原来目标比)	
Goal	1	Result	2
总结规律	Insight		3
经验 & 规律 (不要轻易下结论)		分析原因	
行动计划		成功关键因素 (主观/客观)	
新举措:		失败根本原因 (主观/客观)	
叫停:			
继续:			

同学们, 经过 20 多个任务的实操演练, 相信各个团队在 Android 应用开发的技能上都有了质的提高, 团队间的配合也越来越默契了, 有的团队已经提早进入了成熟阶段! 祝愿各位同学能把所学的知识技能应用到将来的工作岗位上, 创造属于你的辉煌!

6.1.5 任务扩展(Extend)

课后练习: 到友盟和 App Annie 的官方网站了解其他关于消息推送和社会化分享的功能。

附件1 《智能农业实战项目需求说明书》模板

目 录

- 1 引言
 - 1.1 目的
 - 1.2 定义
 - 1.3 参考资料
- 2 软件总体概述
 - 2.1 软件标识
 - 2.1.1 项目名称
 - 2.1.2 产品标识
 - 2.2 软件描述
 - 2.2.1 系统属性
 - 2.2.2 开发背景
 - 2.2.3 系统功能
- 3 具体需求
 - 3.1 功能需求
 - 3.1.1 登录界面

1 引言

1.1 目的

- 定义软件总体要求,作为用户、软件开发人员以及其他干系人之间沟通的基础;
- 描述功能要求、性能要求、用户和系统的接口要求等内容,作为软件开发人员进行软件结构设计和编码的基础;
- 作为软件总体测试的依据。

1.2 定义

甲方:联想教育与培训事业部。
乙方:xxx有限公司。

1.3 参考资料

《招标书》

《投标书》
 《项目管理制度》
 《软件工程国家标准汇编》

2 软件总体概述

2.1 软件标识

2.1.1 项目名称

项目名称：移动智能农业管理系统

项目编号：

产品范围：

2.1.2 产品标识

产品名称：

产品简称：

版本号：1.0

2.2 软件描述

2.2.1 系统属性

该系统不仅可以作为高职高专学生的综合实训课程，而且可以作为期末考试的考核题目。

2.2.2 开发背景

软硬结合、用户体验至上已成为移动互联网未来发展趋势，移动互联网时代开发的产品，最终反映到终端上，则是以用户为中心的设计，在给用户带来丰富的内容和服务的同时，更重要的是给用户提供良好的体验。为了培养高职高专学生的软硬结合的产品开发意识和用户体验设计能力，激发高职高专学生对移动互联网应用软件开发领域的学习和研究兴趣，提高其软件编程能力和职业素养，使学生顺应移动互联网的发展趋势，为中国移动互联网产业发展输送新鲜血液，需要开发一套用于高职高专学生的移动智能农业管理系统。

2.2.3 系统功能

序号	功能名称	优先级	简要描述
1	系统注册登录		
1.1	系统注册	中	注册一个新用户
1.2	系统登录	中	注册用户登录系统
1.3	忘记密码	低	根据注册信息找回密码

(续表)

序号	功能名称	优先级	简要描述
2	远程监测		
2.1	空气温度	高	以实时曲线图的方式显示空气温度
2.2	空气湿度	高	以实时曲线图的方式显示空气湿度
2.3	土壤湿度	高	以实时曲线图的方式显示土壤湿度
2.4	光照强度	高	以实时曲线图的方式显示光照强度
2.5	二氧化碳浓度	高	以实时曲线图的方式显示二氧化碳浓度
3	系统设置		
3.1	空气温度阀值设置	高	设置空气温度的最小值和最大值
3.2	空气湿度阀值设置	高	设置空气湿度的最小值和最大值
3.3	土壤湿度阀值设置	高	设置土壤湿度的最小值和最大值
3.4	光照强度阀值设置	高	设置光照强度的最小值和最大值
3.5	二氧化碳浓度阀值设置	高	设置二氧化碳浓度的最小值和最大值
4	远程控制		
4.1	打开和关闭风扇	高	当空气温度、空气湿度或二氧化碳浓度不在设置的区间时，就会打开风扇
4.2	打开和关闭 LED 灯	高	当光照强度不在设置的区间时，就会打开 LED 灯
4.3	打开和关闭水泵	高	当土壤湿度不在设置的区间时，就会打开水泵

3 具体需求

3.1 功能需求

联想实验机作为数据采集器，和沙盘的单片机板通过串口相连，用于采集各种传感器的数据；同时，联想实验机作为服务器，和客户端 App 进行通讯，将采集到的传感器数据实时发送到客户端 App 进行显示，并根据设置的阀值打开和关闭沙盘中的控制设备，包括：喷水器、LED 灯、风扇。

3.1.1 登录界面



附件2 《智能农业实战项目概要设计说明书》模板

目 录

1 简介

1.1 目的

1.2 范围

1.2.1 软件名称

1.2.2 名称解释

1.2.3 服务器端软件功能

1.2.4 客户端软件功能

2 总体设计

2.1 整体框架

2.2 服务器端模块设计

2.3 客户端模块设计

3 协议接口设计

3.1 HTTP——json 实例

3.2 HTTP——soap 实例

3.3 SOCKET 实例

1 简介

1.1 目的

本文的目的是为了描述智能农业的设计思想,主要包括:

1. 介绍智能农业的总体设计思路。
2. 描述智能农业服务器端与客户端的子模块划分。
3. 描述智能农业客户端与服务器端之间的网络接口。

本文档经过评审修改并基线化以后,将作为编码阶段的输入以便后续阶段工作的正常展开。该文档将为系统测试计划以及测试用例写作提供必要的相关参考信息。该文档也可供软件组其他模块软件开发人员、测试人员参考之用。

1.2 范围

1.2.1 软件名称

智能农业。

1.2.2 名称解释

缩略语	英文全名	中文解释
JSON	JavaScript Object Notation	JSON 是一种轻量级的数据交换格式,完全独立于语言的文本格式
SOAP	Simple Object Access Protocol	简单对象访问协议是交换数据的一种协议规范,是一种轻量的、简单的、基于 XML 的协议,它被设计成在 WEB 上交换结构化的和固化的信息

1.2.3 服务器端软件功能

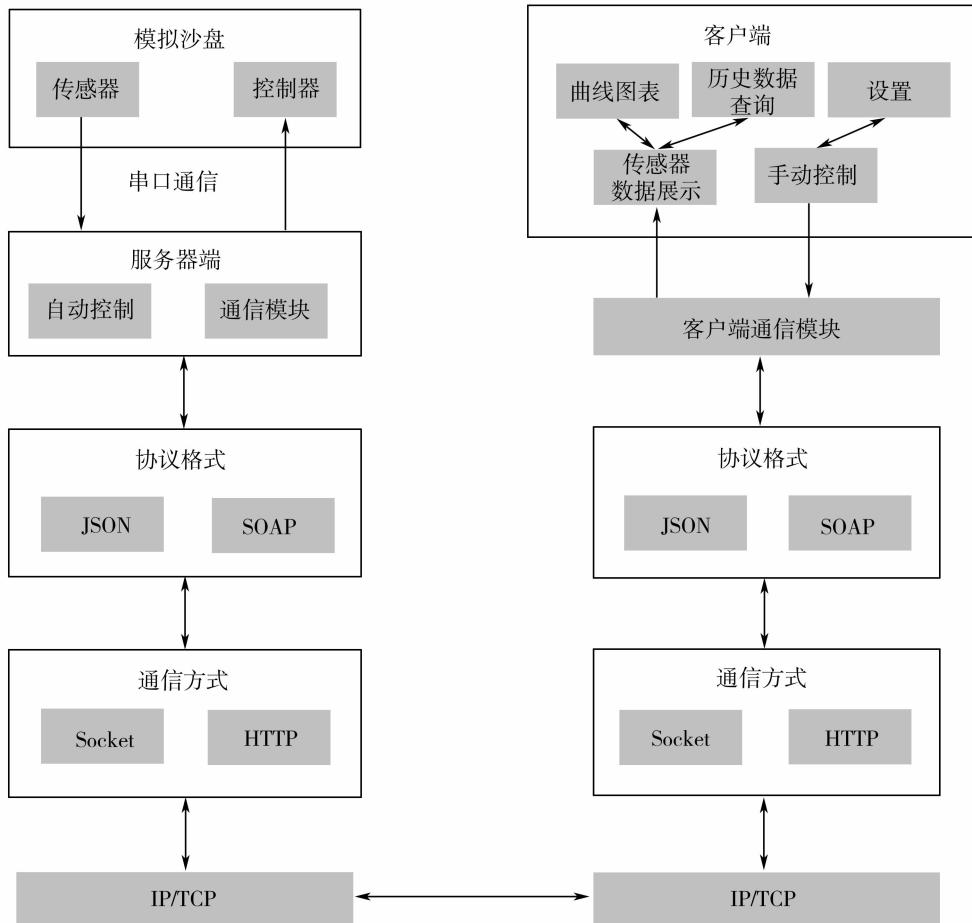
1. 传感器数据采集
2. 服务器启动界面
3. 协议模块(JSON 和 SOAP 协议)
4. 设置串口端口
5. 自动控制逻辑
6. 通信模块(HTTP 或 SOCKET)
7. 模拟数据
8. 传感器阀值设置
9. 日志模块

1.2.4 客户端软件功能

1. 用户登录
2. 用户注册
3. 用户找回密码
4. 设置服务器端 IP
5. 实时数据图表显示界面
6. 历史数据展示界面
7. 手动控制
8. 系统语言设置
9. 自动手动控制切换
10. 传感器阀值设置
11. 通信模块(HTTP 或 SOCKET)
12. 协议请求模块(JSON 和 SOAP 协议)
13. 警告消息推送(通知栏里显示)

2 总体设计

2.1 整体框架



1. 模拟沙盘: 模拟沙盘对真实环境中的各传感器的功能进行抽象。抽取其中六个传感单元(光照、空气温湿度、土壤温湿度、CO₂)和四个控制单元(风扇、水泵、灯、报警器)进行模拟, 模拟的数据通过串口线连接到服务器端。

2. 服务器端: 服务器端主要负责管理系统的用户(登录、注册和找回密码)、定时采集当前数据, 负责适配与各种客户端之间的协议连接(HTTP 和 SOCKET), 并将客户端需要的数据通过协议接口反馈至客户端。还能接收来自客户端的控制信息, 对四个控制单元(风扇、水泵、灯、报警器)进行控制。

3. 客户端: 客户端负责对系统的数据和状态进行显示和控制, 包括实时数据曲线图表、历史数据展示和手动控制。

4. 服务器端——自动控制,在设置成自动控制模式以后,服务器端会根据传感器的当前数据和阀值空间判断是否要控制 4 个控制单元。

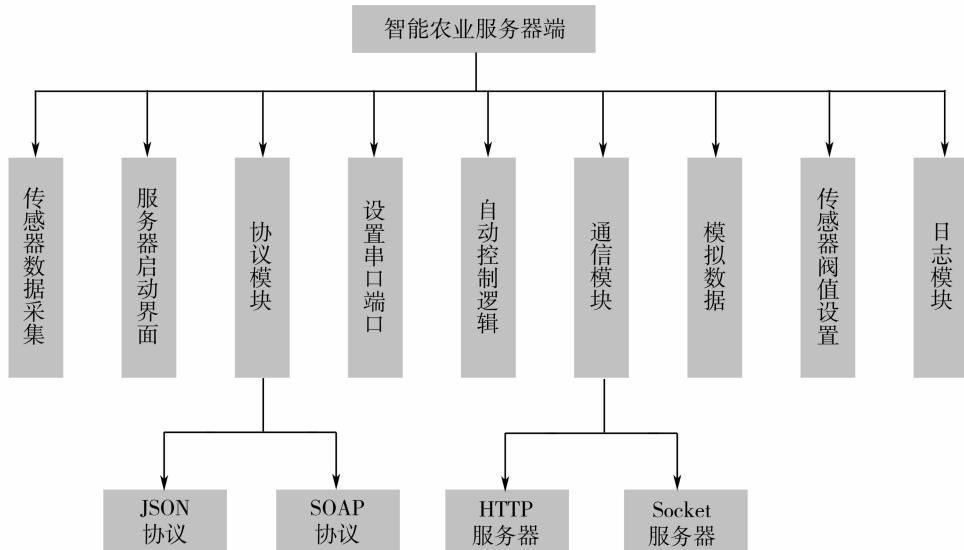
5. 客户端——手动控制,不管当前是否在自动控制模式下,客户端都可以进行手动控制 4 个控制单元。

6. 通信协议格式,包括 JSON 和 SOAP 两种数据格式协议,与具体的通信方式无关,既可以通过 HTTP 协议通信,也可以通过 SOCKET 进行通信。一条完整的请求协议要包括协议类型、请求动作名称和请求 body 三组数据。在 HTTP 通信方式下,协议类型和请求动作名称存放在 URL 中,请求 body 放在 HTTP body 里面,具体见 jason 协议文档和 soap 协议文档。在 SOCKET 通信方式下,三组数据存储在一个 json 报文里,如: {'type:'jason','action': 'login', body: 'BASE64[协议请求参数]'}, 具体见 socket 协议文档。

7. 通信方式:包括 HTTP 和 SOCKET 两种通信方式,HTTP 服务器端端口是 8890,采用 POST 方式。SOCKET 服务器端口是 8891。

8. 日志模块:服务端实现日志记录功能,用于记录客户端的信息,操作类型及操作时间等相关信息,服务端实现日志的浏览功能(在服务端主界面单击浏览日志按钮,打开日志浏览界面)。

2.2 服务器端模块设计



1. 传感器数据采集:服务器通过串口定时不断地采集模拟沙盒数据,一秒钟采集一次。

2. 用户模块:服务器提供了用户注册、登录和找回密码等接口,用户数据存储在后台数据库。当用户要找回密码时,服务器端会将该用户名和密码发送到该用户邮箱。

3. 服务器端主界面:该界面展示了服务器的 IP 地址,并提供了设置串口端口、开启

服务器和开启模拟数据的功能。

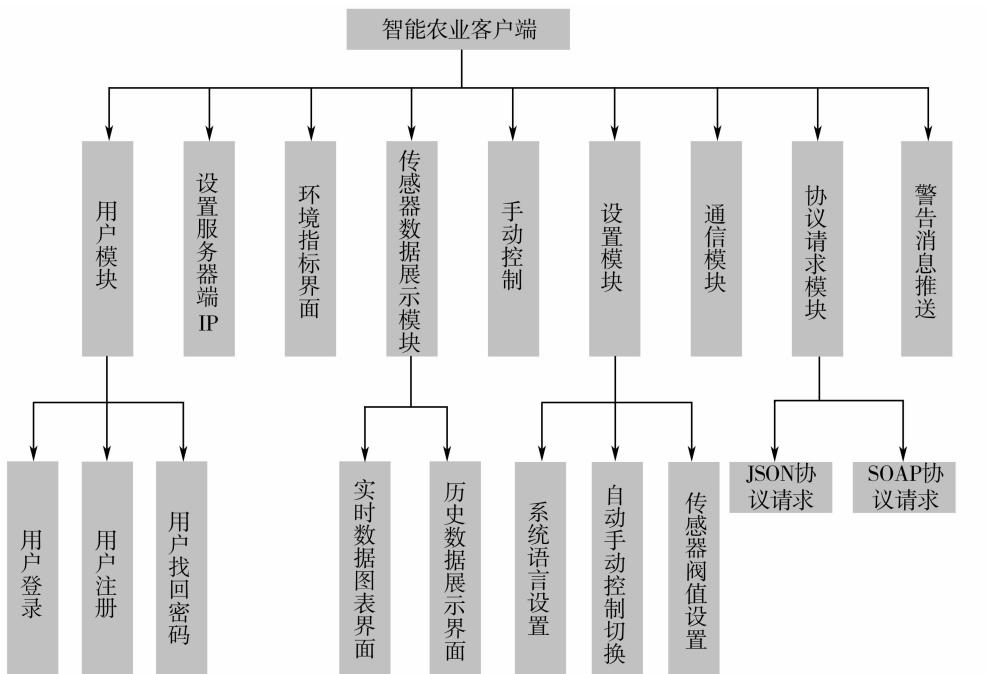
4. 协议模块,服务器能同时支撑 JSON 和 SOAP 两种数据格式协议。协议详情见 json 协议文档和 soap 协议文档。

5. 通信模块,服务器端拥有 HTTP 和 SOCKET 两个服务器,端口分别是 8890 和 8891。

6. 模拟数据,服务器端可以采用随机数的方式随机产生各个传感器的数据值,在没有模拟沙盒的情况下特别有用。

7. 传感器阀值设置,服务器端存储了每个传感器的最大值和最小值,客户端可以进行设置,系统会根据这些阀值来判断是否进入告警状态,在自动控制模式下可以直接控制 4 个控制单元。

2.3 客户端模块设计



1. 用户模块:功能包括用户注册、用户登录和找回密码。

2. 管理员权限:当用户是 admin 时,则拥有管理员权限,可以进行手动控制和各个传感器的阀值设置。其他用户则只能获取和展示传感器数据,不能进行控制。

3. 环境指标界面:该界面实时展示了所有传感器的当前数据值,并能根据每个传感器的阀值判断是否应该告警,如果需要告警则显示红色背景,正常则是绿色背景。

4. 实时数据图表界面:该界面通过实时图表的方式展示了每个传感器的数据值,共有 7 个图表,可以左右滑动进行切换。每隔 1 秒取一次实时刷新一次,将数组内的数据向前移动一位,第一位丢弃,最后一位空出,并将读取到的数据置于数组的最后一位。

5. 历史数据展示界面:客户端将读取到的传感器数据存储到数据库,用户可以通过

选择传感器类型和时间段,从数据库中读取历史数据,然后以图表的方式进行展示,支持左右滚动和缩放。

6. 手动控制界面:系统在手动控制模式下,用户可以控制 4 个控制单元(风扇,水泵,灯,报警器)。

7. 系统语言设置:客户端提供中英文两套资源文件,并在系统设置功能里提供设置开关,当用户点击设置时会跳转到系统语言设置界面,当选择为中文时,系统取用中文的资源文件。

8. 自动手动控制切换:系统提供两种控制方式,一种是自动控制,另一种是手动控制,该功能在系统设置功能中进行设置。

9. 传感器阀值设置:每个传感器的最大值和最小值,客户端可以进行设置,系统会根据这些阀值来判断是否进入告警状态。

10. 通信模块:客户端可以选择使用 HTTP 方式或 SOCKET 方式与服务器进行通信,HTTP 服务器端口是 8890,SOCKET 服务器端口是 8891。

11. 协议请求模块:客户端可以选择使用 JSON 或者 SOAP 数据格式协议与服务端进行交互。

12. 用户引导界面:在第一次进入客户端应用时,会展示用户引导界面,共三页,点击最后一页可以进入用户登录界面。

13. 警告消息推送:当传感器出现任一阀值告警时,客户的通知栏都会收到告警消息。当该条告警不存在或手工清除之后,客户端通知栏内的对应告警信息也可以自动清除。

3 协议接口设计

客户端与服务器端的通信方式采用问答模式,即客户端先发出提问请求,服务器接收以后,根据查询结果给出回应。

客户端一条完整的请求协议应该包括协议类型、请求动作名称和请求 body 三组数据。在 HTTP 通信方式下,协议类型和请求动作名称存放在 URL 中,请求 body 放在 HTTP body 里面。

3.1 HTTP——json 实例

比如在 HTTP 通信方式下,用户登录的 JSON 请求内容应该是:

```
Url : http://192.168.1.119:8890/type/jason/action/login
Body: {'username': 'admin', 'password': 'admin'},
```

其中 type/jason 指明了 json 协议类型,action/login 指明了用户登录请求动作名称,{'username': 'admin', 'password': 'admin'} 即请求 body。此时服务端的回应可以是:{'result': 'ok'} 表示成功,或者{'result': 'failed'} 表示失败。

3.2 HTTP——soap 实例

比如在 HTTP 通信方式下,用户登录的 SOAP 请求内容应该是:

```
Url : http://192.168.1.119:8890/type/soap/action/login
Body:
<? xml version="1.0" encoding="utf-8"? >
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <username>admin</username>
        <password>admin</password>
    </soap:Body>
</soap:Envelope>
```

其中 type/jason 指明了 soap 协议类型,action/login 指明了用户登录请求动作名称,

```
<? xml version="1.0" encoding="utf-8"? >
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <username>admin</username>
        <password>admin</password>
    </soap:Body>
</soap:Envelope>即请求 body。
```

此时服务端的回应可以是:

```
<? xml version=\\"1.0\\" encoding=\\"utf-8\\"? >
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <result>ok</result>
    </soap:Body>
</soap:Envelope>表示登录成功。
<? xml version=\\"1.0\\" encoding=\\"utf-8\\"? >
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
    <soap:Body>
        <result>failed</result>
    </soap:Body>
```

</soap:Envelope>表示登录失败。

3.3 SOCKET 实例

比如在 SOCKET 通信方式下,用户登录的 JSON 请求内容应该是:

```
{'type': 'jason', 'action': 'login', 'body': 'BASE64[协议请求参数]'}
```

其中协议请求参数请见 jason 和 soap 协议文档中的用户登录协议。

此时服务端的回应是:

协议返回内容 = BASE64 解码[socket 接收内容]

其中协议返回内容见 jason 和 soap 协议文档中的用户登录协议的返回内容。

附件3 《智能农业实战项目详细设计说明书》模板

目 录

- 1 服务器端详细设计
 - 1.1 服务器端主界面
 - 1.1.1 功能描述
 - 1.1.2 实现方案
 - 1.1.3 相关类
 - 1.1.4 核心函数
 - 2 客户端详细设计
 - 2.1 用户登录界面
 - 2.1.1 功能描述
 - 2.1.2 实现方案
 - 2.1.3 相关类
 - 2.1.4 核心函数

1 服务器端详细设计

1.1 服务器端主界面

1.1.1 功能描述

展示了服务器的 IP 地址，并提供了设置串口端口、开启服务器和开启模拟数据的功能。

1.1.2 实现方案

1. 通过系统 WifiManager 管理器获取本地 IP。

```
WifiManager wifiManager = (WifiManager) getSystemService(WIFI_SERVICE);  
WifiInfo wifiInfo = wifiManager.getConnectionInfo();  
int ipAddress = wifiInfo.getIpAddress();
```

2. 启动时判定 HTTP service 和串口 service 是否已经启动，并根据相关结果设置开关状态。

1.1.3 相关类

MainActivity 服务器主窗体类。

1.1.4 核心函数

获取本地 IP 地址：

函数原型	private String getLocalIpAddress() throws UnknownHostException
功能描述	获取本地 IP 地址
Input 输入参数	N/A
Output 输出参数	N/A
Return 返回值	N/A
Exception 抛出异常	UnknownHostException

实现描述：

```
private String getLocalIpAddress() throws UnknownHostException
{
    WifiManager wifiManager = (WifiManager) getSystemService(WIFI_SERVICE);
    WifiInfo wifiInfo = wifiManager.getConnectionInfo();
    String IP;
    int ipAddress = wifiInfo.getIpAddress();
    IP = intToIp(ipAddress);
    return IP;
}
```

2 客户端详细设计

2.1 用户登录界面

2.1.1 功能描述

1. 登录界面包括信息有：用户名，密码，是否记住密码，忘记密码，取消，登录。
2. 该功能需要对输入的信息进行过滤，对不满足要求的输入在客户端进行处理。
3. 对满足要求的数据，封装后，发往服务器进行鉴权。

2.1.2 实现方案

1. 使用 android 的配置类 SharedPreferences 保存用户名和密码。
2. 如果用户名和密码不符合要求，则提示用户。
3. 采用 HTTP 或 SOCKET 的通信方式向服务器发送请求。

2.1.3 相关类

1. UserLoginActivity，用户登录界面。
2. LoginRequest，用户登录请求类。

2.1.4 核心函数

1. 开始登录函数

函数原型	private void userLoginEntry()
功能描述	用户开始登录函数
Input 输入参数	N/A
Output 输出参数	N/A
Return 返回值	N/A
Exception 抛出异常	N/A

函数实现：

```
//先检查用户名和密码
if(username.equals("")){
    showAlertDialog(getString(R.string.prompt),
    getString(R.string.username_not_empty));
    return;
}
if(password.equals ""){
    showAlertDialog(getString(R.string.prompt), getString(R.string.password_not_empty));
    return;
}
//先保存用户名和密码
mApp.setUsername(username);
```

参考资料

1. (美) Project Management Institute 项目管理协会. 项目管理知识体系指南(PMBOK 指南)(第 5 版). 许江林,译. 北京:电子工业出版社,2013.
2. 张斌. 题解《PMBOK 指南》PMP 备考指南(第 3 版). 北京:电子工业出版社,2014.
3. (美)Ken Schwaber Jeff Sutherland . 30 天软件开发:告别瀑布拥抱敏捷. 王军,李麟德,译. 北京:人民邮电出版社,2014.
4. (美)Venkat Subramaniam Andy Hunt. 高效程序员的 45 个习惯:敏捷开发修炼之道. 钱安川,郑柯,译. 北京:人民邮电出版社,2010.
5. 邹欣. 构建之法——现代软件工程. 北京:人民邮电出版社,2014.
6. (英)Ian Sommerville. 软件工程(第 9 版). 程成,译. 北京:机械工业出版社,2011.
7. 联想复盘方法论,联想集团内部资料
8. 尚学堂科技 PPT,网络公开资料